

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

«На правах рукопису»
УДК 004.05

«До захисту допущено»

Завідувач кафедри СПСКС

_____ В.П.Тарасенко
(підпис) (ініціали, прізвище)
“ ” _____ 2018р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 123 Комп'ютерна інженерія
Системне програмування

на тему: Спосіб перетворення GL-моделей за критерієм мінімуму
складності реберних функцій

Виконала: студентка II курсу, групи КВ-62м
(шифр групи)

Малишева Мілена Олегівна
(прізвище, ім'я, по батькові) _____ (підпис)

Науковий керівник доц., к.т.н. Романкевич В.О.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____ (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) _____ (підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ – 2018 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність 123 Комп'ютерна інженерія

Системне програмування

ЗАТВЕРДЖУЮ

Завідувач кафедри СПСКС

_____ В.П.Тарасенко
(підпис) (ініціали, прізвище)

« ____ » _____ 2018р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Малишевій Мілені Олегівні
(прізвище, ім'я, по батькові)

1. Тема дисертації: СПОСІБ ПЕРЕТВОРЕННЯ GL-МОДЕЛЕЙ ЗА КРИТЕРІЄМ МІНІМУМУ СКЛАДНОСТІ РЕБЕРНИХ ФУНКЦІЙ, науковий керівник дисертації Романкевич Віталій Олексійович, к.т.н., доцент, затверджені наказом по університету від «22» березня 2018 р. №986-с

2. Термін подання студентом дисертації _____ травня 2018 р.

3. Об'єкт дослідження: відмовостійкі багатопроцесорні системи та моделі, що відображають їх поведінку в потоці відмов.

4. Предмет дослідження: спосіб перетворення базових GL-моделей в небазові, що дозволяє отримати простіші реберні функції.

5. Перелік завдань, які потрібно розробити:

- провести аналіз підходів до обчислення показників надійності ВБС;
- дослідити існуючі моделі ВБС та проаналізувати можливість їх використання для відображення поведінки небазових систем в потоці відмов;
- проаналізувати способи створення базових GL-моделей;
- обґрунтувати вибір критерію оптимізації GL-моделей;

- запропонувати методику поділу елементів системи на групи в алгоритмі послідовного поділу на групи для спрощення перетворення базових GL-моделей в небазові;
- розробити та проаналізувати спосіб перетворення GL-моделей в небазові за критерієм мінімуму складності реберних функцій.

6. Перелік ілюстративного матеріалу:

- алгоритм послідовного поділу елементів системи на групи для формування GL-моделей;
- блок-схема алгоритму послідовного поділу елементів системи на групи для формування базових GL-моделей;
- відображення підвищення ступеня відмовостійкості системи за допомогою додавання конститuentи;
- розподіл відмов за підмножинами з потужністю $m+k$;
- запропонований спосіб модифікації GL-моделей для відображення підвищення ступеня відмовостійкості системи;
- запропонований спосіб відображення підвищення ступеня відмовостійкості для блокування векторів з $m+1$ нульовими компонентами.

7. Перелік публікацій:

- тези доповіді “Спосіб перетворення GL-моделей за критерієм мінімуму складності реберних функцій”, ПМК, 2018;
- тези доповіді “Method of modification of base graph-logical models”, SAIT, 2018.

8. Дата видачі завдання 5 вересня 2016 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Грунтовне ознайомлення з предметною галуззю	27.11.2016	
2.	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	09.04.2017	
3.	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	25.05.2017	
4.	Проведення наукового дослідження; робота над другим розділом магістерської дисертації	10.10.2017	
5.	Проведення наукового дослідження; робота над третім розділом магістерської дисертації	08.12.2017	
6.	Проведення наукового дослідження; підготовка матеріалів доповіді на	04.02.2018	

	конференції ПМК-2018		
7.	Завершення роботи над основною частиною магістерської дисертації; підготовка ілюстративного матеріалу	04.04.2018	
8.	Оформлення текстової і графічної частини магістерської дисертації	18.04.2018	
9	Попередній розгляд магістерської дисертації на кафедрі	26.04.2018	

Студент

(підпис)

М.О. Малишева

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

В.О. Романкевич

(ініціали, прізвище)

РЕФЕРАТ

Актуальність теми. З кожним роком галузь застосування відмовостійких багатопроцесорних систем розширюється. Через те, що такі системи обробляють одночасно інформацію з багатьох модулів, вони мають надзвичайно складну структуру та велику кількість елементів. Ці системи часто використовують для керування важливими об'єктами, виведення з ладу яких може спричинити тяжкі соціальні та економічні наслідки. Саме тому визначення рівня надійності для них є надзвичайно складним та важливим завданням.

Один з найважливіших показників надійності – ймовірність безвідмовної роботи обчислюють за допомогою виконання статистичних експериментів над спеціальними моделями. Зрозуміло, що простота таких моделей впливає на кількість експериментів, які можна виконати над ними за фіксовану кількість часу, та точність обчислення ймовірності безвідмовної роботи відповідно.

Серед існуючих моделей відмовостійких багатопроцесорних систем особливо привабливими через свою простоту виглядають так звані GL-моделі, що мають форму неорієнтованого зв'язного графу, кожному ребру якого відповідає булева реберна функція.

Існує декілька алгоритмів побудови вказаних реберних функцій для моделей базових систем. Проте більшість відмовостійких багатопроцесорних систем в наш час є небазовими, саме тому актуальним є завдання створення способів перетворення GL-моделей для відображення поведінки небазових системи в потоці відмов, що дозволяють отримати простіші реберні функції, ніж за умови стандартних способів перетворення (наприклад, модифікація функцій з додавання конститuentи).

Об'єктом дослідження є відмовостійкі багатопроцесорні системи та моделі, що відображають їх поведінку в потоці відмов.

Предметом дослідження є спосіб перетворення базових GL-моделей в небазові, що дозволяє отримати прості реберні функції.

Мета роботи: аналіз існуючих алгоритмів модифікації базових GL-моделей відмовостійких багатопроцесорних систем в небазові, створення способу перетворення GL-моделей за критерієм мінімуму складності реберних функцій, визначення умови доцільного використання способу перетворення GL-моделей за критерієм мінімуму складності реберних функцій.

Наукова новизна полягає в наступному:

1. Запропоновано спосіб поділу модулів на підмножини для алгоритму послідовного поділу елементів системи на групи в залежності від набування відповідними компонентами нульового значення в векторах, що необхідно блокувати.
2. Запропоновано спосіб перетворення GL-моделей в небазові за критерієм мінімуму складності реберних функцій.
3. Проаналізовано використання запропонованого способу для відображення підвищення ступеню відмовостійкості систем на GL-моделях.

Практична цінність отриманих в роботі результатів полягає в тому, що запропонований спосіб дозволяє отримати простіші реберні функції, ніж за умови використання базового способу відображення підвищення ступеню відмовостійкості з додаванням конститuentи. Таким чином над отриманими моделями можна виконати більшу кількість експериментів за фіксований час та обчислити рівень надійності системи з більшою точністю.

Апробація роботи. Основні положення і результати роботи були представлені на:

- X науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2018-1 (Київ, 21-23 березня 2018 р.)
- XX міжнародній конференції SAIT (Київ, 21 – 24 травня 2018 р.).

Публікації. За результатами проведених досліджень опубліковано 2 наукові роботи у вигляді двох тез доповідей у збірниках матеріалів конференцій.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, трьох розділів, висновків по кожному розділу та загальних висновків по роботі, списку використаної літератури та додатків.

У вступі зроблено оцінку сучасного стану обраного напрямки досліджень, показано наукову новизну отриманих результатів і практичну цінність роботи.

У першому розділі проаналізовано поняття надійності та показників надійності, розглянуто основні підходи до їх обчислення, наведено класифікацію відмов, та обґрунтовано необхідність використання моделей відмовостійких багатопроцесорних систем.

У другому розділі розглянуто моделі, що можуть бути використані для відображення поведінки відмовостійких багатопроцесорних систем в потоці відмов, надано поняття GL-моделей, проведено аналіз існуючих алгоритмів створення базових GL-моделей.

У третьому розділі досліджено алгоритм послідовного поділу елементів системи на групи для створення базових GL-моделей, розглянуто способи перетворення базових GL-моделей в небазові за допомогою модифікації реберних функцій та модифікації графу, проаналізовано існуючі критерії оптимізації GL-моделей, запропоновано та проаналізовано новий спосіб перетворення базових GL-моделей в небазові, що дозволяє отримати простіші реберні функції, ніж за допомогою використання стандартного способу додавання константи для відображення підвищення ступеню відмовостійкості систем на моделях.

У висновках представлені результати проведеної роботи.

Робота представлена на 81 аркушах, містить посилання на список використаних літературних джерел.

Ключові слова: надійність ВБР, відмовостійкість, GL-модель, побудова GL-моделей, перетворення GL-моделей.

РЕФЕРАТ

Актуальность темы. С каждым годом область применения отказоустойчивых многопроцессорных систем расширяется. Так как такие системы часто должны обрабатывать одновременно информацию со многих модулей, они имеют чрезвычайно сложную структуру и большое количество элементов. Эти системы часто используют для управления важными объектами, вывод из строя которых может вызвать тяжелые социальные и экономические последствия. Именно поэтому определение уровня надежности для них является чрезвычайно сложной и важной задачей.

Один из важнейших показателей надежность – вероятность безотказной работы вычисляют с помощью выполнения статистических экспериментов над специальными моделями. Понятно, что простота таких моделей влияет на количество экспериментов, которые можно выполнить над ними за фиксированное количество времени, и на точность вычисления вероятности безотказной работы соответственно.

Среди существующих моделей отказоустойчивых многопроцессорных систем особенно привлекательными из-за своей простоты являются так называемые GL-модели, имеющие форму неориентированного связного графа, каждому ребру которого соответствует булева реберная функция.

Существует несколько алгоритмов построения таких реберных функций для GL-моделей базовых систем. Однако большинство отказоустойчивых многопроцессорных систем в наше время является небазовым, поэтому актуальной является задача создания способов преобразования GL-моделей для отображения поведения небазовых системы в потоке отказов, позволяющие получить более простые реберные функций, чем при использовании стандартных способов преобразования (например, модификации функций с добавлением конститутенты).

Объектом исследования отказоустойчивые многопроцессорные системы и модели, отражающие их поведение в потоке отказов.

Предметом исследования является способ преобразования базовых GL-моделей в небазовые, который позволяет получить простые реберные функции.

Цель работы: анализ существующих алгоритмов модификации базовых GL-моделей отказоустойчивых многопроцессорных систем, создание способа преобразования GL-моделей по критерию минимума сложности реберных функций, определения условия целесообразного использования способа преобразования GL-моделей по критерию минимума сложности реберных функций.

Научная новизна заключается в следующем:

1. Предложен способ разделения модулей на подмножества для алгоритма последовательного разделения элементов системы на группы в зависимости от приобретения соответствующими компонентами нулевого значения в векторах, которые необходимо блокировать.
2. Предложен способ преобразования GL-моделей в небазовые по критерию минимума сложности реберных функций.
3. Проанализировано использование предложенного способа для отображения повышения степени отказоустойчивости систем на GL-моделях.

Практическая ценность полученных в работе результатов заключается в том, что предложенный способ позволяет получить более простые реберные функции, чем при использовании базового способа отображения повышения степени отказоустойчивости с добавлением конститутенты. Таким образом над полученными моделями можно выполнить большее количество экспериментов за фиксированное время и вычислить уровень надежности системы с большей точностью.

Апробация работы. Основные положения и результаты работы были представлены на:

- X научной конференции магистрантов и аспирантов «Прикладная математика и компьютеринг» ПМК-2018-1 (Киев, 21-23 марта 2018)

- XX международной конференции SAIT (Киев, 21 - 24 мая 2018).

Публикации. По результатам проведенных исследований опубликовано 2 научные работы в виде двух тезисов докладов в сборниках материалов конференций.

Структура и объем работы. Магистерская диссертация состоит из введения, трех глав, заключения по каждому разделу и общих выводов по работе, списка использованной литературы и приложений.

Во *введении* произведена оценка современного состояния избранного направления исследований, показано научную новизну полученных результатов и практическую ценность работы.

В *первом разделе* проанализированы понятия надежности и показателей надежности, рассмотрены основные подходы к их вычислению, приведена классификация отказов и обоснована необходимость использования моделей отказоустойчивых многопроцессорных систем.

Во *втором разделе* рассмотрены модели, которые могут быть использованы для отображения поведения отказоустойчивых многопроцессорных систем в потоке отказов, приведено понятие GL-моделей, проведен анализ существующих алгоритмов создания базовых GL-моделей.

В *третьем разделе* исследован алгоритм последовательного деления на элементов системы группы для создания базовых GL-моделей, рассмотрены способы преобразования базовых GL-моделей в небазовые с помощью модификации реберных функций и модификации графа, проанализированы существующие критерии оптимизации GL-моделей, предложен и проанализирован новый способ преобразования базовых GL-моделей в небазовые, который позволяет получить более простые булевы функции, чем с помощью использования стандартного метода добавления констант для отображения повышения степени отказоустойчивости систем на моделях.

В *выводах* представлены результаты проведенной работы.

Работа представлена на 81 листах, содержит ссылки на список использованных литературных источников.

Ключевые слова: надежность ВБР, отказоустойчивость, GL-модель, построение GL-моделей, преобразование GL-моделей.

ABSTRACT

Relevance of the topic. Application field of fault-tolerant multiprocessor systems expands every year. Since such systems often need to process information from many modules simultaneously, they have an extremely complex structure and a large number of elements. These systems are often used to manage important objects, the failure of which can lead to severe social and economic consequences. That is why determining the level of reliability for them is extremely difficult and important task.

One of the most important characteristics of reliability – failure-free operation probability is calculated by performing statistical experiments on special models. The simplicity of such models obviously affects the number of experiments that can be performed on the model for a fixed period of time, and the accuracy of calculating failure-free operation probability, respectively.

Among the existing models of fault-tolerant multiprocessor systems, the so-called GL-models are particularly attractive because of their simplicity. A GL-model has the form of an undirected connected graph, with the Boolean edge function corresponding to each edge of the graph.

There are several algorithms for generation of edge functions for GL-models of basic systems. However, most of the multiprocessor systems are non-basic nowadays, so the relevant task is to find the ways of transformation GL-models to reflect the behavior of non-basic systems in the failure flow, which allow to obtain simpler edge functions than using standard methods of transformation (for example, modification of functions with the addition of constituent).

Research object is fault-tolerant multiprocessor system and a model that reflect its behavior in the failure flow.

Research subject is the method of transformation basic GL-models to non-basic ones, which allows to obtain simple edge functions.

Goals of the thesis: analysis of the existing methods for transformation of basic GL-models into non-basic ones, creation the method for transformation

GL-models based on the criterion of minimum edge functions complexity, determining the conditions for the expedient use of the method by the criterion of minimum edge functions complexity.

The scientific novelty lies in the following:

1. The way of dividing modules into subsets for the algorithm of sequential elements division into groups, depending on the amount of times corresponding components get the zero value in the vectors that need to be blocked, is proposed.
2. The method for GL-models transformation into non-basic ones based on the criterion of minimum complexity is proposed.
3. The use of the proposed method for reflection the increasing in the fault tolerance degree on GL-models is analyzed.

Practical relevance of the results obtained in the work is that the proposed method makes it possible to get simpler edge functions than while using the classic method with the constituent's addition. In this way, more experiments can be performed on the model in the fixed period of time and the level of reliability of the system can be calculated more accurately.

Approbation of the thesis. The main statements and results of the work were presented and discussed at:

- the Xth scientific conference of undergraduate and post-graduate students "Applied Mathematics and Computing" PMK-2018-1 (Kyiv, March 21-23, 2018);
- the XX-th International Conference SAIT 2018(Kyiv, May 21-24, 2018).

Publications. According to the research results 2 scientific works were published in the form of two abstracts in the conference proceedings.

Structure and size of the thesis. The master's thesis consists of an introduction, three sections, conclusion for each section and general conclusions for the thesis and the list of references.

In the introduction the current state of the selected research area was described, the scientific novelty of the results obtained and the practical relevance of the work were shown.

In the first section the concepts of reliability and reliability characteristics were analyzed, the main approaches to their calculation were considered, the failure classification was presented and the necessity of using models of fault-tolerant multiprocessor systems was substantiated.

In the second section models that can be used to describe fault-tolerant multiprocessor systems in a failure flow were considered, the concept of GL-models was given and the existing algorithms for creating basic GL-models were analyzed.

In the third section the algorithm of sequential division into groups for creating basic GL-models was described, the methods of transformation basic GL-models to non-basic ones by means of edge functions and graph modification were considered, existing optimization criteria for GL-models are analyzed, a new way of transformation of basic GL-models into non-basic ones, which allows to obtain simpler Boolean functions than with the standard method of adding constitutions, was proposed and analyzed.

The conclusion contains the results of the work done.

The thesis has 81 pages, contains references to the list of literature used.

Keywords: reliability of, fault tolerance, GL-model, GL-models construction, GL-models transformation.

ЗМІСТ

ВСТУП.....	5
1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ МАГІСТЕРСЬКОЇ РОБОТИ	7
1.1. Основні поняття	7
1.2. Стани ВБС.....	12
1.3. Характеристики відмовостійких рішень.....	15
1.4. Управління надійністю	16
1.5. Діагностування та контроль ВБС	17
1.6. Підвищення рівня надійності систем.....	20
1.7. Підходи до обчислення кількісних показників надійності ВБС	22
Висновки розділу 1	25
2. ПОБУДОВА МОДЕЛЕЙ ВБС	26
2.1. Загальні підходи до формування моделей ВБС	26
2.1.1 Засновані на паралельних та послідовних структурах моделі ..	27
2.1.2 Моделі k-out-of-n.....	33
2.2. GL-моделі.....	36
2.3. Побудова GL-моделей базових ВБС	40
Висновки розділу 2	43
3. РОЗРОБЛЕНИЙ МЕТОД ПЕРЕТВОРЕННЯ GL-МОДЕЛЕЙ.....	44
3.1. Алгоритм послідовного поділу елементів системи на групи для формування GL-моделей.....	44
3.2. Побудова GL-моделей для небазових ВБС	52
3.3. Мінімізація GL-моделей.....	58
3.4. Критерії оптимальності GL-моделей	65
3.5. Спосіб початкового розбиття елементів системи групи для алгоритму послідовного поділу на групи для спрощення перетворення базових моделей в небазові	68
3.6. Запропонований спосіб відображення підвищення ступеня відмовостійкості	72

3.7. Аналіз способу перетворення GL-моделей в небазові шляхом відображення підвищення ступеня відмовостійкості.	75
Висновки розділу 3	77
ВИСНОВКИ.....	78
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	80
ДОДАТКИ	
Додаток 1. Копії графічних матеріалів	
Плакати 1. Алгоритм послідовного поділу елементів системи на групи для формування GL-моделей	
Плакати 2. Блок-схема алгоритму послідовного поділу елементів системи на групи для формування базових GL-моделей	
Плакати 3. Відображення підвищення ступеня відмовостійкості системи за допомогою додавання конститuentи	
Плакати 4. Розподіл відмов за підмножинами з потужністю $m+k$	
Плакати 5. Запропонований спосіб модифікації GL-моделей для відображення підвищення ступеня відмовостійкості системи	
Плакати 6. Аналіз запропонованого способу відображення підвищення ступеня відмовостійкості для блокування векторів з $m+1$ нульовими компонентами	

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ВБС – відмовостійка багатопроцесорна система;

ЙБР – ймовірність безвідмовної роботи;

GL-модель – графо-логічна модель;

КС – комп'ютерна система;

ДНФ – диз'юнктивна нормальна форма;

КНФ – кон'юнктивна нормальна форма.

ВСТУП

В наш час в багатьох сферах використовуються так звані відмовостійкі багатопроцесорні системи (ВБС). Особливу увагу варто приділити їх використанню при керуванні важливими об'єктами зі складною структурою. Такі системи містять дуже велику кількість модулів, дані з яких необхідно обробляти одночасно та визначати послідовність дій на їх основі. Прикладами таких систем можуть слугувати атомні електростанції, літаки, супутники.

На етапі їх проектування необхідно оцінювати дуже важливу характеристику таких систем – ймовірність безвідмовної роботи (ЙБР), яка визначається в процесі виконання експериментів над моделями, які створюються для відображення поведінки ВБС в потоці відмов.

Такі моделі можуть мати значно простішу структуру, ніж структура реальних ВБС, що дає можливість провести більше експериментів над ними та відповідно точніше обчислити показники надійності систем, поведінку яких ці моделі відображають.

Існує багато різних варіантів структури моделей. Найбільш простими моделями, наприклад, є моделі з паралельно-послідовними зв'язками, структура яких в цілому відповідає структурі ВБС. Проте зрозуміло, що такі системи не дають можливість обчислити показники надійності системи з великою кількістю елементів за прийнятний час.

Тому в якості альтернативи було запропоновано так звані графо-логічні, або GL-моделі, які дозволяють обчислити ЙБР (ймовірність безвідмовної роботи) швидко та з використанням прийнятних обчислювальних ресурсів. GL-модель є графом довільної структури, ребрам якого ставлять у відповідність булеві реберні функції, які залежать від стану модулів реальної системи [1]. В такому випадку вважається, що система відмовила, якщо граф відповідної моделі втратив зв'язність.

На теперішній час існує багато підходів до формування моделей базових систем (систем, які втрачають роботоздатність за появи фіксованої кількості відмов на елементах). Але зрозуміло, що реальні системи дуже часто є небазовими. Для перетворення базових GL-моделей в небазові існує методика

додавання конститuentи, яка проте не враховує особливостей алгоритмів побудови базових систем та потребує подальшої мінімізації реберних функцій.

Тому головною метою даної магістерської дисертації є знаходження способу перетворення GL-моделей в небазові на основі алгоритму послідовного поділу на групи за критерієм мінімуму складності реберних функцій.

1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ МАГІСТЕРСЬКОЇ РОБОТИ

1.1. Основні поняття

Однією з основних задач етапу проектування реконфігурованих ВБС [2-3] є забезпечення їх довготривалого надійного функціонування. Реконфігурованою відмовостійкою системою називають систему, яка здатна до пошуку, налізу та усуненню відмов модулів для збереження свого роботоздатного стану. Приклад концептуального дизайну відмовостійкої комп'ютерної системи зображено на рис. 1.1.

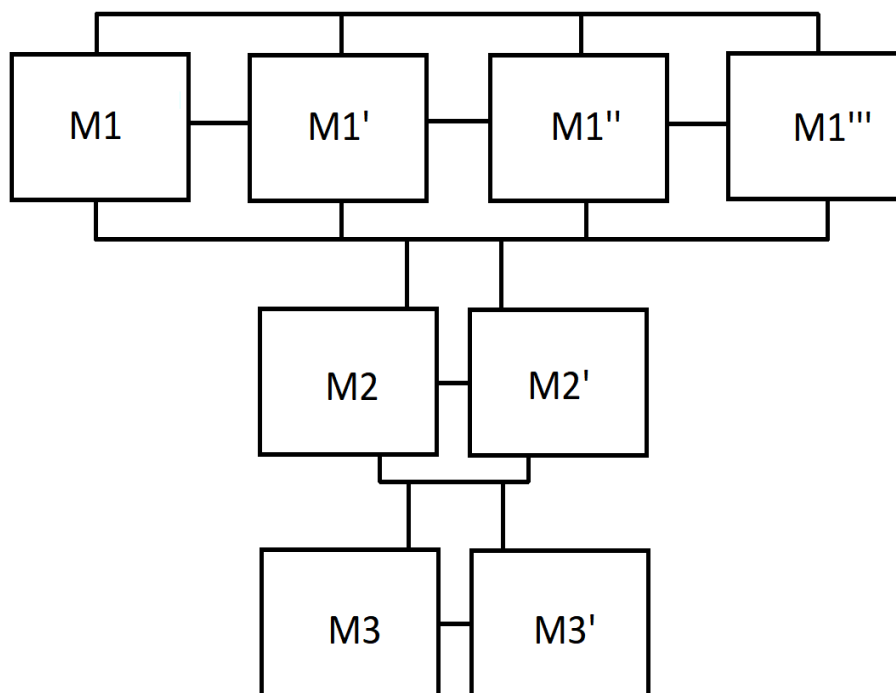


Рис. 1.1. Приклад концептуального дизайну відмовостійкої комп'ютерної системи

Підвищена увага до проблем надійності комп'ютерних систем обумовлена наступними факторами [2]:

- підвищення складності пристроїв та поява складних систем;
- повільне зростання надійності окремих компонентів системи в порівнянні з ростом їх кількості;
- підвищення важливості функцій, що виконують комп'ютерні системи, і, як наслідок, підвищення вимог до показників надійності таких систем;

– ускладнення умов експлуатації комп'ютерних систем.

Відмовостійкість визначається кількістю послідовних одиничних відмов елементів системи, поява яких не впливає на втрату роботоздатності системою в цілому.

За способами реалізації відмовостійкість прийнято поділяти на активну та пасивну.

Активна відмовостійкість базується на процесах виявлення відмови, локалізації відмови та реконфігурації системи. Виявлення відмов виконується за допомогою використання засобів контролю, локалізація – за допомогою засобів діагностування. Після цього відмови усуваються автоматичною реконфігурацією системи.

Пасивна відмовостійкість базується на властивості системи втрачати свої функціональні можливості у випадку відмови окремих її компонентів. Таким чином пасивна відмовостійкість обов'язково пов'язана зі збільшенням кількості апаратури в декілька разів. Пасивну відмовостійкість використовують або у випадку, коли система, що проектується, не повинна втрачати роботоздатність навіть на короткий проміжок часу, або ж коли необхідно забезпечити відмовостійкість для найбільш важливих елементів системи.

За умови використання активної відмовостійкості апаратні засоби використовуються більш економно, ніж за умови використання пасивної відмовостійкості. Однак активна відмовостійкість не гарантує роботоздатний стан системи в кожний момент часу, бо вона пов'язана з деякими затримками для відновлення роботи системи після відмови а також з можливою втратою необхідних для роботи даних.

Активна відмовостійкість реалізується тільки в багатопроцесорних системах (із загальною пам'яттю, загальною шиною, матричної, кільцевої або іншої структури). У той же час застосування пасивної відмовостійкості гарантує практично безупинну роботу КС (комп'ютерної системи) і збереження всієї інформації.

Розглянемо поняття «надійності».

В наш час дослідження надійності виконується в декількох основних напрямках[3]:

- статистична теорія надійності займається визначенням показників надійності на основі спеціального тестування об'єкту або на основі даних о практичній експлуатації об'єктів;
- математична теорія надійності досліджує можливість використання математичних методів та моделей для передбачення, аналізу показників надійності та забезпечення прийняттого рівня надійності технічних пристроїв;
- фізична теорія надійності зосереджується на фізичних причинах відмов технічних об'єктів (наприклад, вплив старіння і зносу матеріалів або інших зовнішніх впливів).

Поняття «надійності» використовують для позначення властивості об'єкту зберігати в часі роботоздатний стан в заданих документацією умовах використання, технічного обслуговування та транспортування [4].

Роботоздатний стан в цьому випадку передбачає можливість виконання об'єктом оговорених функцій з параметрами, які вказані в його технічній специфікації.

Надійність необхідно розглядати не тільки з точки зору апаратних засобів, але й з точки зору програмного забезпечення.

Надійність є складною комплексною властивістю та може в залежності від конкретного об'єкту та контексту складатися з більш простих підвластивостей (рис 1.2.). Відповідно і показники надійності поділяють на декілька груп [3]:

- показники безвідмовності;
- показники ремонтпридатності;
- показники зберігаємості;
- показники довговічності.

Безвідмовністю називають здатність об'єкта бути в роботозданому стані за вказаних в документації умов.



Рис. 1.2. Взаємозв'язок надійності з іншими властивостями об'єктів

Математично безвідмовність можна описати за допомогою виразу:

$$R(t) = \Pr\{T > t\} = \int_t^{\infty} f(x)dx \quad (1.1)$$

В формулі (1.1) $f(x)$ – функція щільності ймовірності відмови.

До показників безвідмовності відносять:

- ймовірність безвідмовної роботи (ЙБР), обчисленню якої в подальшому буде присвячено увагу;
- інтенсивність відмов;
- середній наробіток на відмову та ін.

Ремонтпридатністю називають властивість системи, що характеризує її пристосованість до швидкого пошуку причин відмов та технічного обслуговування і ремонту.

До показників ремонтпридатності відносять:

- середній час відновлення роботоздатного стану;
- ймовірність відновлення роботоздатного стану та ін.

Довговічністю називають здатність системи або об'єкта до довготривалого збереження роботоздатного стану за умови дотримання умов використання, зазначених в специфікації. Довговічність визначається моральним або фізичним (технічним) зносом.

До показників довговічності відносять:

- максимальний час експлуатації за розрахунками;
- середній ресурс;
- призначений ресурс;
- час до першого капітального ремонту;
- середній час між капітальними ремонтами та ін.

Зберігаємістю називають здатність об'єкту або системи зберігати на потрібному рівні вказані технічні параметри під час зберігання або фізичного транспортування.

До показників зберігаємості відносять:

- гамма-відсотковий час зберігаємості;
- середній час зберігаємості;
- ймовірність відмови під час зберігання;
- інтенсивність відмов під час зберігання та ін.

Для кількісної оцінки декількох підвластивостей надійності також використовують різноманітні комплексні показники.

Класифікація факторів, що впливають на значення показників надійності, зображена на рис.1.3.



Рис. 1.3. Класифікація факторів, що впливають на значення показників надійності

Кількісна оцінка параметрів надійності може виконуватися для вирішення питань:

- порівняння поточного рівня надійності системи з рівнем надійності, вказаним в специфікації;
- розробка вимог до надійності;
- покращення логічної структури системи з точки зору можливого підвищення показників рівня надійності з одночасним прийнятним збільшенням вартості побудови такої системи;
- вибір оптимальних технічних характеристик складових системи з точки зору прийнятного для замовника збільшення вартості загальної побудови системи;
- порівняння та аналіз можливих засобів підвищення різних показників надійності.

1.2. Стани ВБС

Розглянемо більш детально можливі стани системи та їх взаємозв'язки (рис. 1.4.)[2].

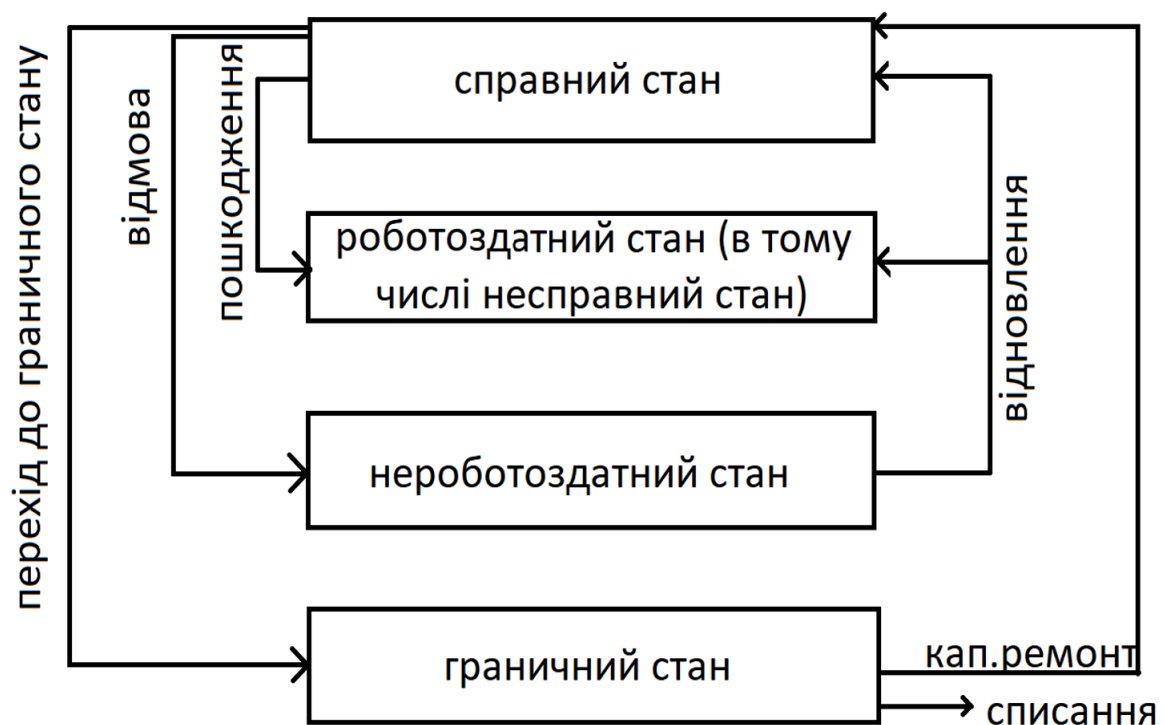


Рис. 1.4. Стани системи

Справний стан – це стан об’єкту, в якому він відповідає всім вимогам технічної документації.

Несправний стан – це стан об’єкту, в якому він не відповідає хоча б одній вимозі технічної документації.

Роботоздатний стан – це стан об’єкту, в якому значення всіх параметрів, що характеризують здатність об’єкту виконувати свої функції, відповідають зазначеним в технічній документації значенням.

Нероботоздатний стан – це стан об’єкту, в якому значення хоча б одного параметру, що характеризує здатність об’єкту виконувати свої функції, не відповідає зазначеному в технічній документації значенню.

Граничний стан – це стан об’єкту, в якому його подальша експлуатація неприпустима, або ж відновлення його роботоздатного стану є неможливим або недоцільним.

Роботозданість системи може бути порушеною у випадку появи відмов. У зв’язку з тим, що під відмовою часто розуміють дещо синонімічне іншим понять теорії надійності (наприклад поняттю «пошкодження»), важливим є визначити поняття відмов та навести їх детальну класифікацію.

Відмова – поняття теорії надійності, що в загальному випадку позначає порушення роботоздатності системи, за появи якого вказана система вже не може виконувати всі оговорені в технічній специфікації функції в повному об’ємі.

Наслідками відмов можуть бути:

- необхідність дострокового ремонту;
- зниження показників продуктивності;
- збільшення витрат на ремонт в майбутньому у випадку ігнорування проблеми.

Відмови класифікують за наступними ознаками:

- за часовим характером [3]:
 - поступові, за яких зміна параметрів об’єкту відбувається поступово в часі (рис 1.5);

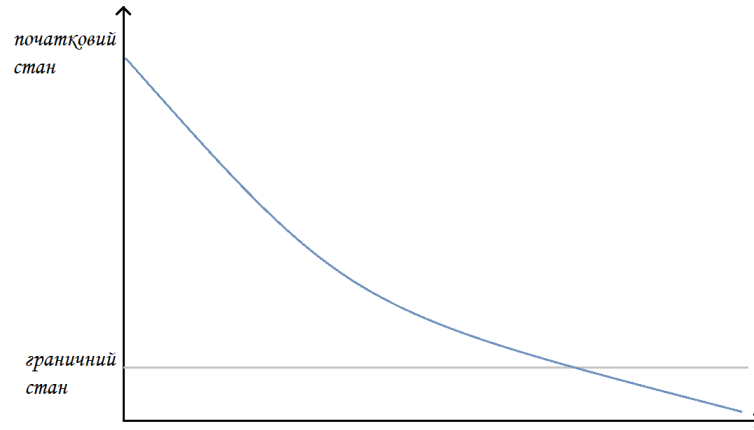


Рис. 1.5. Система за появи поступової відмови

- раптові, за яких стан об'єкту змінюється стрибкоподібно (рис 1.6);

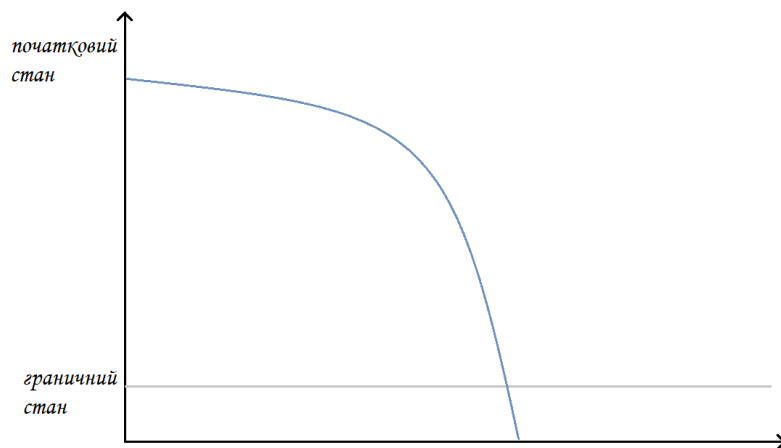


Рис. 1.6. Система за появи раптової відмови

- за моментом виникнення:
 - ранні;
 - пізні.
- за наявності зв'язку з появою відмов на інших модулях:
 - залежні;
 - незалежні;
- за впливом на роботоздатність системи:
 - часткові, за яких системи може виконувати свої функції, проте не повністю;
 - повні, за яких системи вже не може виконувати всі функції;
- за наслідками:
 - звичайні;

- аварійні, що призводять до руйнування самого об'єкту або ж до негативних наслідків для оточуючого середовища;
- за частотою появи:
 - систематичні;
 - випадкові;
- за характером зовнішнього прояву:
 - приховані, які потребують для свого виявлення застосування спеціальних технічних засобів;
 - явні.

1.3. Характеристики відмовостійких рішень

Незважаючи на очевидні переваги використання відмовостійких рішень, існують також і недоліки, які обов'язково потрібно розглянути.

1. Ціна. І активна, і пасивна відмовостійкість збільшуює загальну вартість системи.
2. Складність діагностування. Для деяких вкрай важливих відмовостійких систем, таких, наприклад, як ядерний реактор, немає простого шляху, щоб упевнитися, що запасні частини знаходяться в робочому стані.
3. Зменшення пріоритету виправлення несправності. І хоча цей пункт є доволі суб'єктивним, навіть якщо оператор знає про наявність проблеми, існування відмовостійкості системи зменшить важливість виправлення проблеми.
4. Проблеми з виявленням багатокomпонентних несправностей. Відмовостійкість одного компонента може заважати виявленню несправностей в іншому. Наприклад, відмовостійкість частини В може приховати проблему, що виникла в А. Якщо в подальшому частину В буде замінена на менш відмовостійку, то система може раптово відмовити, при цьому буде здаватися, що проблема полягає в новій частині В.
5. Проблеми з виявленням однокомпонентних несправностей.

1.4. Управління надійністю

Управління надійністю складається з аналізу та забезпечення надійності системи.

Аналіз надійності системи виконується за такою послідовністю [3]:

- дослідження функціонування об'єкта;
- відокремлення окремих модулів системи;
- аналіз можливих відмов та зв'язків між ними;
- вибір необхідного рівня надійності для системи.

Забезпечення рівня надійності систем складається з наступних етапів:

- вибір та адаптація до системи засобів діагностування відмов;
- вибір засобів реконфігурації для відновлення роботоздатного стану системи після відмов;
- ітеративне вдосконалення структури системи та окремих її компонентів для досягнення вказаних в технічній специфікації характеристик надійності. Аналіз рівня надійності системи після вдосконалення. Оптимізація структури відповідно до мінімізації витрат на створення системи, якщо можливо.

На етапі проектування розрахунок надійності виконується з метою прогнозування (передбачення) очікуваної надійності проекрованої системи.

Таке прогнозування необхідно для обґрунтування майбутнього проекту, а також для вирішення організаційно-технічних питань:

- кількості запасних елементів;
- способу резервування;
- вибору оптимального варіанту структури;
- режимів роботи;
- глибини і методів контролю;
- умов експлуатації;
- періодичності профілактики.

1.5. Діагностування та контроль ВБС

Виявлення несправностей в ВБС виконується за допомогою наступних [5] апаратно-програмних засобів:

- початковий контроль, що виконується до етапу початку експлуатації ВБС (тобто на стадії виготовлення);
- оперативний контроль, що виконується в процесі роботи ВБС;
- автономний контроль, що виконується під час тимчасових перерв в роботі ВБС;
- контроль засобів діагностування, що виконується оперативно, або з певною періодичністю.

Розглянемо основні моделі, що використовуються для контролю та діагностування відмовостійких багатопроцесорних систем: P -, B -, R - та K -моделі.

В P -моделі процес контролю та діагностування зображується взаємодією пар елементів системи, що складається з елемента, що контролює, та елемента, якого контролюють. В кожній парі роботоздатність елемента, що необхідно контролювати, за умови справного стану елемента, що контролює, визначається правильно та однозначно.

За відмови елемента, що контролює, правильність результатів його оцінки не є достовірною.

Описані взаємодії між елементами ВБС зображають за допомогою графу $G = G(V, E)$, де V є множиною усіх елементів системи v_i , $i = \overline{1, n}$, $v_i \in V$, а E є множиною зв'язків (v_i, v_j) , $v_i, v_j \in V$, що описують контроль елемента v_j елементом v_i . Результат контролю – множина значень $\{a_{ij}\}$, що визначається за правилом 1.2.

$$a_{ij} = \begin{cases} 0, & \text{якщо елемент } v_i \text{ визнає } v_j \text{ роботоздатним} \\ 1, & \text{якщо елемент } v_i \text{ не визнає } v_j \text{ роботоздатним} \end{cases} \quad (1.2)$$

Тобто, якщо елемент v_i знаходиться у справному стані ($v_i = 0$), a_{ij} буде дорівнювати значенню v_j , у зворотньому випадку a_{ij} буде мати довільне значення.

В B -моделі, на відміну від P -моделі, результат контролю $a_{ij} = 0$ вважається недійсним, якщо елементи v_i та v_j відмовили одночасно. Тобто кількість припустимих станів роботоздатності системи за умови використання даної моделі є меншою.

В B -моделі діагностування кожного елементу системи виконується декількома іншими за допомогою апаратних засобів або ж програмних процедур. N елементам системи ставиться у відповідність множина припустимих одиничних несправностей $F_0 = \{f_1, \dots, f_n\}$, яким відповідає множина тестів $J = \{h_1, \dots, h_p\}$. Кожний тест вважається повним для однієї несправності з множини F_0 .

Необхідно також визначити, який тест є повним. Тест $h_j, j = \overline{1, p}$ є повним для несправності $f_i, i = \overline{1, n}$, якщо він успішно завершується за відсутності несправностей з множини F_0 та не проходить за наявності несправності f_i .

За відмови більш, ніж одного елементу системи, є припустимою наявність множини несправностей F (вираз 1.3.).

$$F = \{F_1, \dots, F_k, \dots, F_{2^n}\} \quad (1.3)$$

Кожна несправність з множини 1.3. є підмножиною множини одиничних несправностей.

Через те, що для несправностей з множини F умова повноти тесту може бути порушеною, визначається множина тестів $T(F_k)$ для кожної несправності F_k для оцінки повноти тестів для неї. Тест h_j викривляється за наявності несправності F_k , якщо результат його виконання не є достовірним.

Таке викривлення тестів одиничних несправностей дозволяє враховувати вплив відмов елементів, які беруть участь у перевірці елемента, що розглядається.

Розглянемо типи несправностей, що відокремлюють в рамках використання K -моделей.

Множину атомарних несправностей позначають буквою A , де $A = \{a_1, \dots, a_{|A|}\}$. Ця множина відображає несправності на найбільш низькому

(логічному) рівні функціонування системи. Їх можна описати переліком функціональних елементів або постійними несправностями.

Множину макронесправностей позначають буквою M , де $M = \{m_1, \dots, m_{|M|}\}$. Ці несправності є елементами системи, які визначають глибину пошуку дефекту в системі.

Множину групових несправностей позначають буквою P , де $P = \{p_1, \dots, p_{|P|}\}$. Такі несправності є наслідком виникнення підмножини атомарних несправностей.

Необхідно також визначити взаємозв'язки між розглянутими множинами несправностей.

Для відображення зв'язку між множинами A та P було визначено функцію f_{ap} , що визначає для кожної атомарної несправності в A рівно один груповий дефект. Виникнення групового дефекту таким чином можливо за умови наявності атомарної несправності.

Для відображення зв'язку між множинами A та M було визначено функцію r_{am} , що також позначається у вигляді $a_i r_{am} m_j$, описує зв'язок між елементами множин A та M та визначає, що макронесправність m_j виникає щоразу, коли виникає атомарна несправність a_i з множини A .

Для відображення зв'язку між множинами M та P було визначено функцію r_{mp} . Цей зв'язок зумовлює, що хоча б один з групових дефектів є наслідком макронесправності.

В системах, що використовують для діагностування К-моделі, діагностування виконується за допомогою атомарних тестів з множини $U = \{u_1, \dots, u_i, \dots, u_{|U|}\}$, макротестів з множини $T = \{h_1, \dots, h_j, \dots, h_{|T|}\}$ та вхідних значень множини $K = \{k_1, \dots, k_{|K|}\}$. Макротест в даному випадку є множиною атомарних тестів.

В якості висновку можна сказати, що Р- та В-моделі в основному розглядають результати взаємодії елементів ВБС між собою, в той час як в Р- та К-моделях більш розглядається функціональна структура системи та використовуються можливості програмних засобів.

Розглянемо способи діагностування в ВБС. Пошук нероботоздатних елементів системи можна виконувати послідовно з заміною знайденого елементу системи одразу після його знаходження або ж одночасно для всіх несправних елементів.

Кожний з цих способів потребує інформацію про роботоздатність кожного окремого елементу системи, яку можна отримати за допомогою перерахування всіх допустимих поєднань станів роботоздатності елементів. Для даної множини визначаються усі характерні несправності та виконується контроль роботоздатності окремих елементів системи за необхідності.

1.6. Підвищення рівня надійності систем

Завдання підвищення рівня надійності систем може бути вирішеним за допомогою:

- резервування;
- заміна окремих модулів системи на більш надійні;
- вибір структурної схеми системи, що забезпечує більш високий рівень надійності.

Резервування передбачає наявність в системі резервних (надлишкових) елементів, які у випадку відмови основного елементу можуть виконувати його функції.

В теорії надійності виділяють такі види резервування:

- інформаційне;
- апаратне;
- часове.

Апаратне резервування також називають структурним. Воно передбачає використання надлишкових елементів системи. Таким чином в робочий варіант системи, що складається з мінімуму основних елементів, вводяться додаткові (резервні) елементи.

У деяких випадках проводиться навіть дублювання цілої системи один або кілька разів. При цьому надлишкові структурні елементи призначені для того, щоб взяти на себе виконання функцій елементів, які вони замінюють, у разі відмови.

Інформаційне резервування передбачає використання надлишкової інформації. Необхідно також відзначити, що реалізувати інформаційне резервування без додаткових елементів системи часто неможливо.

Часове резервування передбачає використання надлишкового часу для підвищення надійності. Таким чином крім мінімального часу, який необхідний для виконання функцій системою, враховується також деякий запас, який потрібен для відновлення перерваного в результаті відмови функціонування системи.

Структурне резервування може бути реалізовано двома способами:

- загальне резервування, що передбачає резервування цілої системи;
- роздільне (послементне) резервування, що передбачає резервування деяких окремих частин (елементів) системи.

Кратністю резервування називають ступінь резервування (або відношення кількості резервних елементів до кількості основних елементів системи).

Для системи, що складається з п'яти елементів (рис. 1.7.), загальне резервування з кратністю, що дорівнює одиниці, зображено на рис. 1.8., а послементне резервування кожного елемента з кратністю 2 може мати вигляд, як зображено на рис. 1.9.

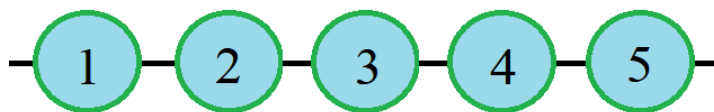


Рис. 1.7. Початкова система

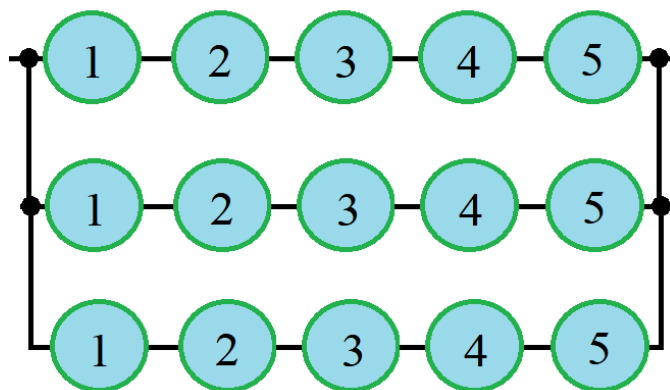


Рис. 1.8. Загальне резервування

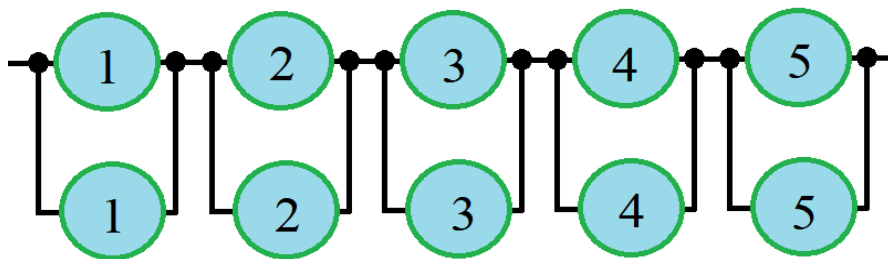


Рис. 1.9. Поелементне резервування

Далі під резервуванням будемо мати на увазі структурне резервування.

В ВБС з великою кількістю елементів нормальною практикою є введення структурного резервування. Однак завжди потрібно знаходити баланс між забезпеченням високої надійності системи та часом, потрібним на діагностування, проектування та підтримку систем з великою кількістю надлишкових елементів.

Потрібно пам'ятати, що основним завданням проектувальника є забезпечення заданого в технічній специфікації рівня надійності, а не забезпечення максимально можливого її рівня. Саме тому актуальним є завдання отримання точного значення показників надійності ВБС (зокрема, ймовірності безвідмовної роботи).

1.7. Підходи до обчислення кількісних показників надійності ВБС

Використання ВБС для керування важливими об'єктами зумовлює те, що розрахунок та оцінка рівня надійності таких систем є надзвичайно актуальним завданням. Цей процес особливо ускладнює те, що такі ВБС складаються з великої кількості елементів, які взаємопов'язані між собою деяким чином.

Існує два основних метода розрахунку рівня надійності ВБС:

- статистичне визначення рівня надійності (на основі обробки результатів експериментів, кількість яких впливає на точність обчислення показників надійності);
- математичне обчислення рівня надійності (за допомогою використання спеціальних математичних моделей, для яких можна сформулювати формули для обчислення параметрів надійності).

Для розрахунку ймовірності безвідмовної роботи систем існує багато методів, але вони або ж ресурсовитратні, або не дають точних результатів.

Наведемо декілька з них:

- метод Белфорда;
- метод послідовно-паралельної редукції;
- метод Рушді;
- метод ланцюгів Маркова;
- метод Барлоу-Хейдмана.

Для вирішення цього завдання також можна використовувати метод прямого перебору, проте зрозуміло, що для великої кількості модулів цей метод буде потребувати дуже багато часу.

Розрахунок показників надійності найчастіше виконують структурними методами, які використовують структурну схему, що повинна відображати взаємозв'язки між модулями системи.

Загальний алгоритм обчислення показників надійності системи зображено на рис. 1.10.

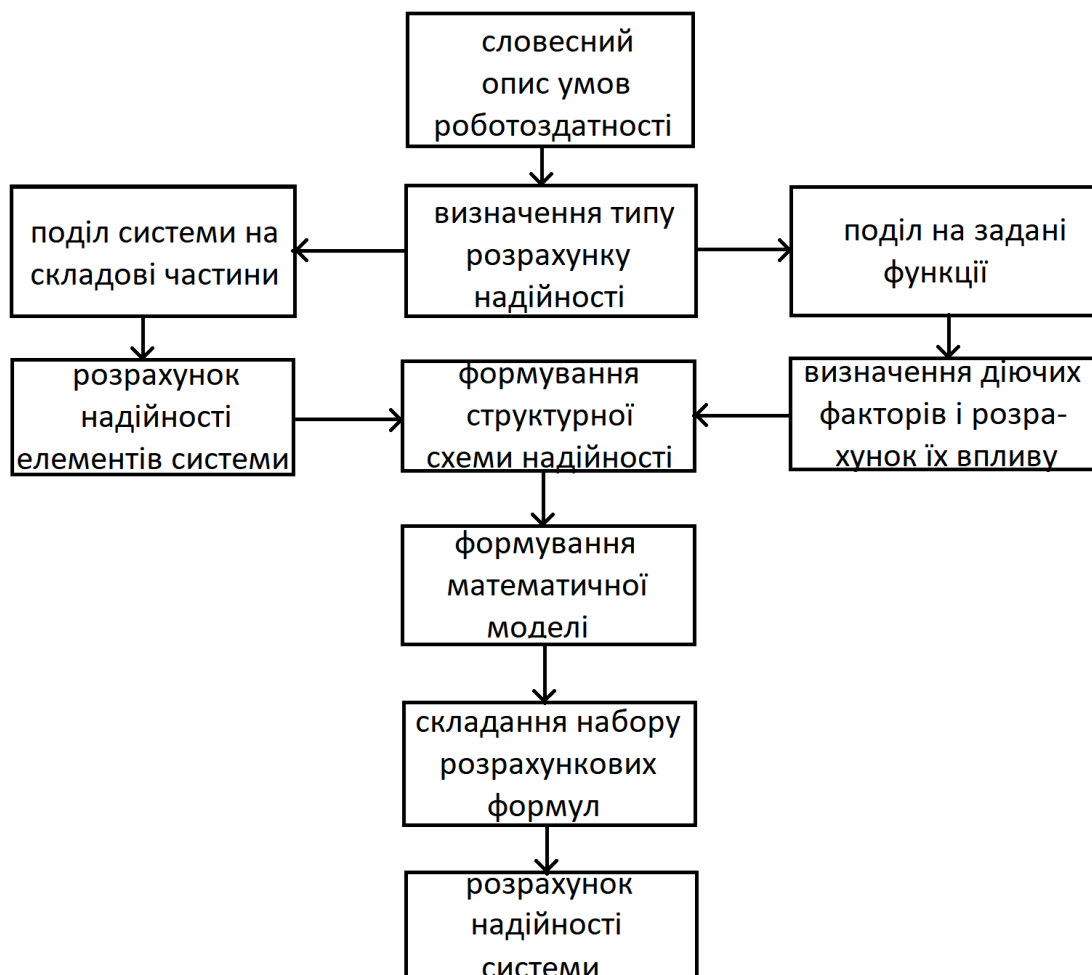


Рис. 1.10. Загальний алгоритм для обчислення показників надійності

Вказана на рисунку математична модель не обов'язково повинна відображати структуру системи, її ціль – опис функціонування об'єкту за появи відмов. Після формування математичної моделі та складання розрахункових формул на її основі виконується розрахунок рівня надійності системи.

Висновки розділу 1

В даному розділі визначено поняття надійності та різних її показників, обчислення яких є дуже важливим завданням під час проектування відмовостійких багатопроцесорних систем. Наведені основні характеристики відмовостійких рішень. Встановлено, що управління надійністю систем складається з обчислення показників надійності та забезпечення їх необхідного рівню. Зазначено, що обчислення показників надійності часто виконується за допомогою спеціальних моделей систем, які повинні відображати функціонування системи за появи відмов на елементах.

2. ПОБУДОВА МОДЕЛЕЙ ВБС

2.1. Загальні підходи до формування моделей ВБС

Відмовостійкі багатопроцесорні системи бувають базові та небазові. Базовою відмовостійкою багатопроцесорною системою називають ВБС, що зберігає роботоздатний стан за появи певної фіксованої кількості відмов.

Тобто, якщо система наприклад втрачає роботоздатність за відмови будь-яких мінімум чотирьох елементів системи, то вона є базовою 3-відмовостійкою системою. Проте якщо існує вектор стану з більш ніж 3 нульовими компонентами, за появи якого система зберігає роботоздатність, то така система називається небазовою.

Базова m -відмовостійка ВБС, що складається з n модулів, позначається m -ВБС, модель такої системи позначають $K(m, n)$ [6].

Як було зазначено раніше, першим етапом для обчислення показників надійності є опис умов роботоздатності, тобто спочатку необхідно сформулювати умови, за яких схема, що розглядається, не здатна виконувати вказані в технічній специфікації функції.

Зрозуміло, що ця умова повинна залежати від станів модулів системи, які в свою чергу можуть знаходитися в роботоздатному стані та стані відмови. Стан модуля системи в подальшому буде позначати функція $x_i(t)$, яка дорівнює 1, якщо i -тий елемент в поточний момент часу знаходиться в роботоздатному стані, та дорівнює нулю, якщо він знаходиться у стані відмови.

Таким чином, якщо система складається з n модулів, тоді існує 2^n станів (векторів), що впливають на загальний стан системи.

Загальний стан системи описує структурна функція, що в теорії надійності позначається $F(x)$. Дана функція може набувати два значення, в залежності від умов, наведених в виразі 2.1.

$$F(X) = \begin{cases} 1, \text{якщо система роботоздатна в стані } X \\ 0, \text{якщо система відмовила} \end{cases} \quad (2.1)$$

З використанням математичного виразу, що описує умови відмови системи, будують математичну модель ВБС, де підключення блоків формує «шлях успіху» системи. Шляхи між портами вводу і виводу проходять через такі

комбінації блоків, які забезпечують коректне відображення функціонування системи на моделі.

В залежності від структури та/або принципів функціонування системи виділяють наступні типові структури моделей:

- моделі на паралельних структурах;
- моделі на послідовних структурах;
- моделі зі змішаною структурою;
- моделі зі складною структурою (наприклад моделі, засновані на радіальній структурі або GL-моделі, які будуть розглянуті в наступних розділах).

2.1.1 Засновані на паралельних та послідовних структурах моделі

Багато ВБС можуть бути описані за допомогою моделей, заснованих на простих послідовних або/та паралельних структурах [7].

Розглянемо розрахунок показників надійності для систем з простим послідовним з'єднанням елементів (або інакше основним з'єднанням елементів) (рис. 2.1.).

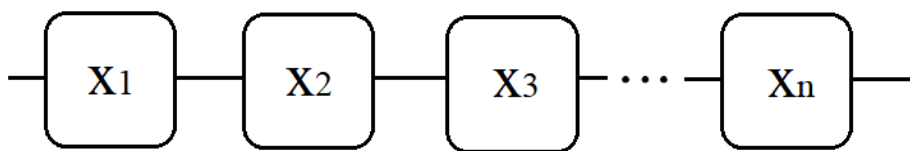


Рис. 2.1. Приклад системи, заснованої на послідовному з'єднанні модулів

Такі моделі використовують, якщо для функціонування системи в цілому необхідно, щоб абсолютно усі блоки системи знаходилися в роботоздатному стані.

Звичайно, фізичне розташування елементів системи може відрізнятися від наведеної моделі, проте головним є те, що відмова бодай одного модулю впливає на втрату роботоздатності цілої системи.

Для такої системи є справедливим вираз 2.2.

$$F(X) = \bigwedge_{i=1}^n x_i \quad (2.2)$$

Після заміни подій на ймовірності виникнення цих подій, ЙБР системи буде дорівнювати:

$$P(F(X)) = \prod_{i=1}^n P_i \quad (2.3)$$

В формулі 2.3 P_i позначає значення ЙБР i -ого модуля системи.

Таким чином для системи з послідовним з'єднанням елементів ЙБР кожного окремого модулю необхідно помножити один на одного для отримання загального показнику ЙБР системи.

Розглянемо розрахунок показників надійності для систем, що мають паралельне з'єднання елементів (рис. 2.2.)

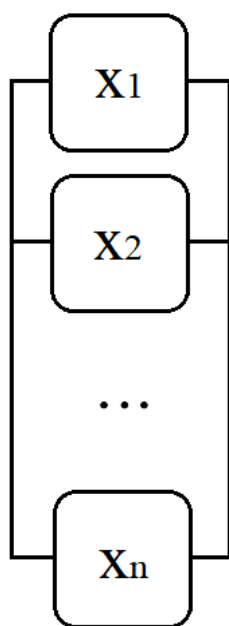


Рис. 2.2. Приклад системи, заснованої на паралельному з'єднанні модулів

Таку моделі використовують, якщо для функціонування системи в цілому необхідно, щоб бодай один блок системи знаходився в роботоздатному стані. Це також означає, що відмова системи в цілому можлива лише у випадку відмови усіх її елементів.

Для такої системи справедливим є вираз 2.4.

$$F(X) = \bigvee_{i=1}^n x_i \quad (2.4)$$

А після перетворення структурної функції 2.4. ЙБР такої системи дорівнює:

$$P(F(X)) = 1 - \prod_{i=1}^n (1 - P_i) \quad (2.5)$$

В цій формулі P_i – ЙБР i -го модуля.

Розглянемо також дві комбінації наведених раніше типів систем: послідовно-паралельна (рис. 2.3.) та паралельно-послідовна, структуру якої зображено на рис. 2.4.

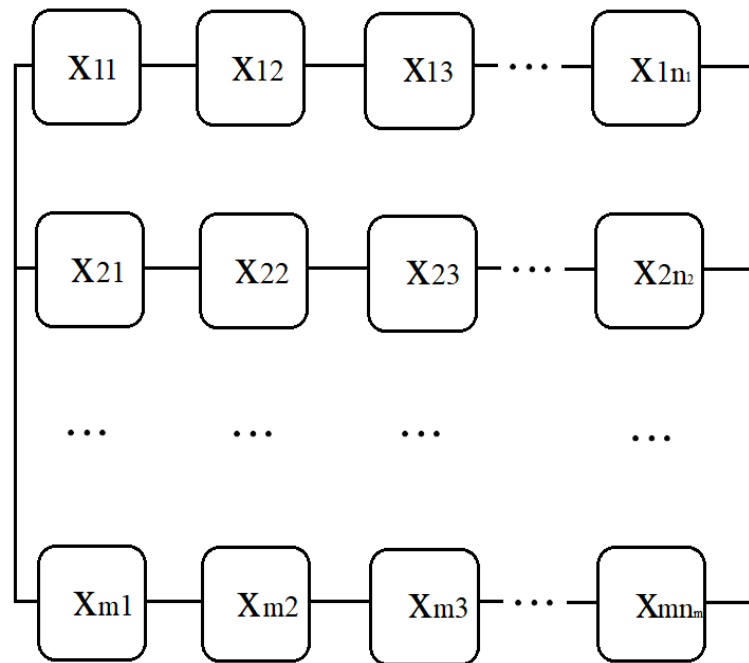


Рис. 2.3. Приклад системи, заснованої на послідовно-паралельному з'єднанні модулів

Для такої системи справедливим є вираз 2.6.

$$F(X) = \bigvee_{j=1}^m \bigwedge_{i=1}^{n_j} x_{ji} \quad (2.6)$$

Таким чином ЙБР системи можна розрахувати за формулою:

$$P(F(X)) = 1 - \prod_{j=1}^m (1 - \prod_{i=1}^{n_j} P_{ji}) \quad (2.7)$$

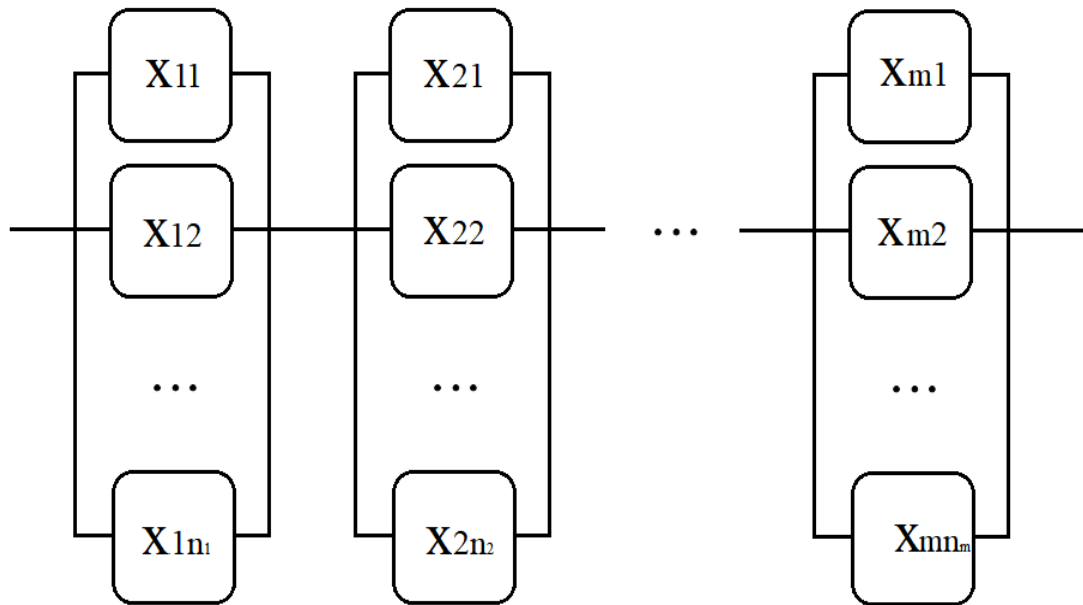


Рис. 2.4. Приклад системи, заснованої на паралельно-послідовному з'єднанні модулів

Для такої паралельно-послідовної системи структурна функція має наступний вигляд:

$$F(X) = \bigwedge_{j=1}^m \bigvee_{i=1}^{n_j} x_{ji} \quad (2.8)$$

А ЙБР паралельно-послідовної системи після перетворень буде дорівнювати:

$$P(F(X)) = \prod_{j=1}^m (1 - \prod_{i=1}^{n_j} (1 - P_{ji})) \quad (2.9)$$

Для моделей, що побудовані на послідовно-паралельних структурах, розрахунок показників надійності є доволі простим, проте зрозуміло, що не у всіх випадках поведінку реальної системи в потоці відмов можливо описати з їх допомогою.

В такому випадку необхідно, якщо це можливо, перетворити складну структурну схему в еквівалентну схему з набором послідовних та паралельних зв'язків або ж сформулювати формули для обчислення показників надійності за допомогою аналізу станів системи.

До методів розрахунку параметрів надійності систем зі складною структурою відносять [2]:

- метод розкладання структури за базовим елементом (базовими елементами);
- метод перетворення трикутника на зірку і назад;
- метод перебору станів.

Розглянемо перші два метода більш детально.

Перетворення схеми трикутника (рис 2.5.) на зірку (рис 2.6.) полягає в заміні складного вузла системи на інший з більш простою конфігурацією, але при цьому необхідно підібрати такі характеристики нового вузла, щоб показники надійності в ньому були еквівалентні показникам надійності у вузлі оригінальної схеми. Для зворотного перетворення зірки в трикутник використовують справедливіми будуть перетворення 2.10.

Проте зрозуміло, що даний метод перетворення дозволяє отримати лише приблизні значення.

$$\begin{aligned} q_{12} &= \sqrt{\frac{q_1 q_2}{q_3}} \\ q_{12} &= \sqrt{\frac{q_2 q_3}{q_1}} \\ q_{12} &= \sqrt{\frac{q_1 q_3}{q_2}} \end{aligned} \quad (2.10)$$

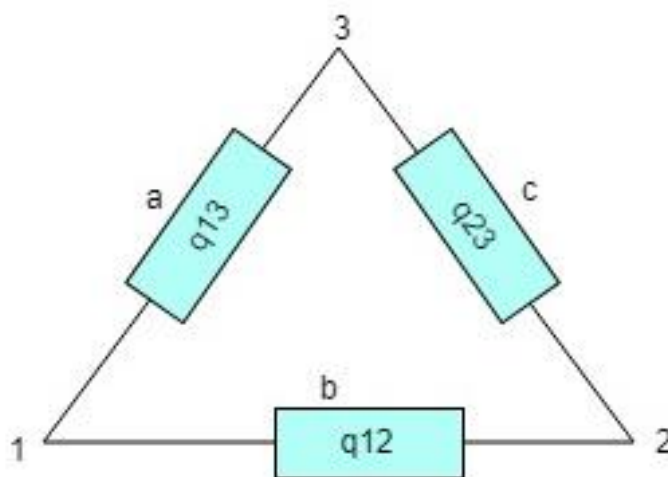


Рис. 2.5. Схема «трикутник»

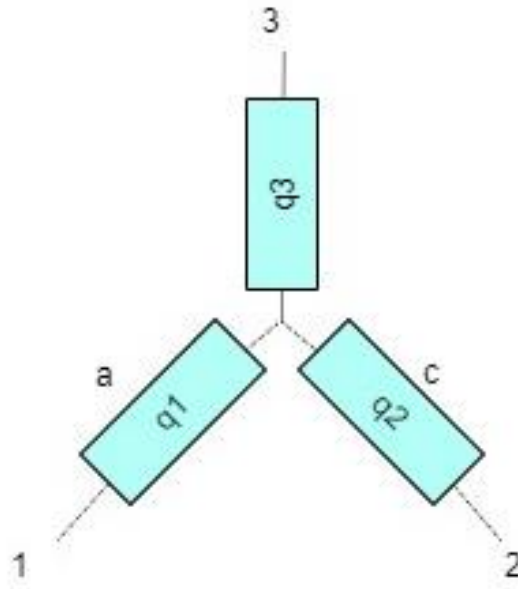


Рис. 2.6. Схема «зірка»

Метод розкладання структури за базовим елементом(базовими елементами) заснований на теоремі ймовірностей несумісних подій (тобто властивості адитивності ймовірностей). В оригінальній структурній схемі обирають один або групу базових елементів, які неможливо перетворити в стандартні паралельно-послідовні структури.

Для таких базових елементів необхідно розглянути обидва варіанти станів:

- елемент відмовив;
- елемент роботоздатний.

Для вказаних подій оригінальну схему надійності перетворюють в дві нові схеми. В першій базові елементи замінюються на коротке замикання ($P(t) = 1$ для $0 \leq t \leq \infty$), в іншій базові елементи замінюються на розрив ланцюга ($P(t) = 0$ для $0 \leq t \leq \infty$).

Потім необхідно розрахувати ймовірності безвідмовної роботи (ЙБР) для вказаних випадків (після спрощення схеми за рахунок видалення елемента вона може бути легко перетворена до стандартних паралельної або послідовної структури) та помножити результат для схеми за умови роботоздатності елемента на ЙБР базових елементів, а результат для схеми за умови відмови елемента на ймовірність втрати роботоздатності базового елемента. Отриманні результати попередніх обчислень необхідно скласти між собою. Обчислена сума

є ймовірністю безвідмовної роботи системи складної структурної схеми надійності.

2.1.2 Моделі k -out-of- n

Часто в реальних задачах систему можна описати за допомогою так званих k -out-of- n моделей [8-9]. Зробимо такі припущення:

- система може знаходитися в стані відмови або в роботоздатному стані;
- кожний компонент може знаходитися в стані відмови або в роботоздатному стані;
- відмови компонентів є незалежними.

Систему, що містить n модулів та зберігає свою роботоздатність тільки за умови роботоздатності щонайменше k елементів називають k -out-of- n : G системою.

Систему, що складається з n структурних елементів та втрачає роботоздатність, якщо k або більше з них втратили роботоздатність, називають k -out-of- n : F системою.

K -out-of- n системами позначають обидва можливі варіанти (F та G системи). Зрозуміло, що будь-яку k -out-of- n можна перетворити до них, проте варто зазначити, що і розглянуті вище послідовні та паралельні системи можуть бути розглянуті як окремі випадки таких систем. Взаємозв'язок між системами зображено на рис. 2.7.

Особливий інтерес для наукового дослідження такі системи мають тому, що вони набули поширення в високотехнологічній сфері промисловості.

Для прикладу ми можемо позначити систему обробки даних, яка складається з семи відеодисплеїв та коректно відображає дані, якщо мінімум чотири відеодисплея роботоздатні, 4 -out-of- 7 : G системою.

Такі k -out-of- n : G системи в літературі часто відображають у вигляді рис. 2.8. Виділені 4 елементи на рисунку позначають, що лише за умови роботоздатності будь-яких 4 елементів система буде знаходитися в роботоздатному стані.

Підтипом звичайних k -out-of- n систем є так звані послідовні k -out-of- n системи [10].

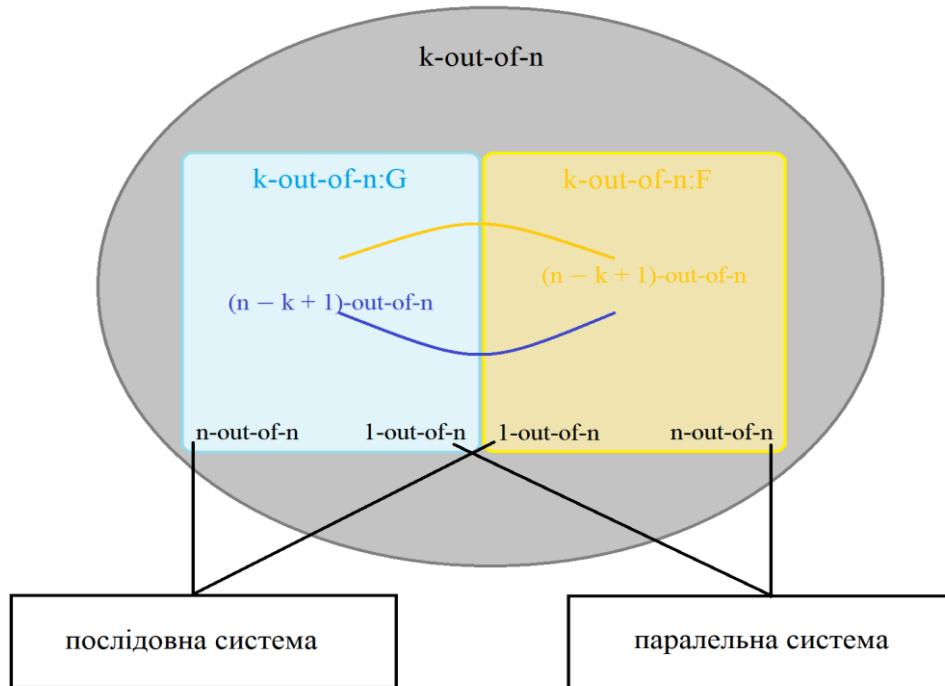


Рис. 2.7. Варіанти k -out-of- n систем

Систему, що складається з n структурних елементів та зберігає свою роботоздатність тільки за умови роботоздатності щонайменше k елементів, розташованих послідовно, називають послідовною k -out-of- n : G системою.

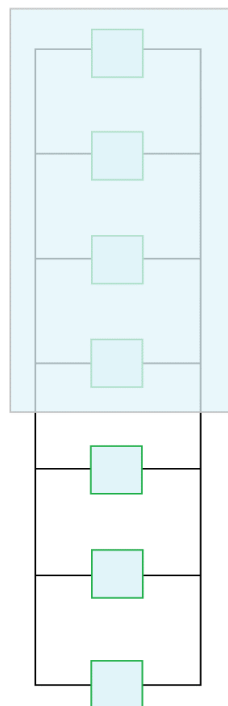


Рис. 2.8. Зображення 4-out-of-7:G системи

Систему, що складається з n структурних елементів та відмовляє тільки за умови відмови щонайменше k елементів, розташованих послідовно, називають послідовною k -out-of- n : F системою.

Прикладом послідовної k -out-of- n : F системи є телекомунікаційна система з n супутниками або наземними станціями, що відмовляє, якщо принаймні дві станції, розташовані послідовно, відмовили. Приклад 3-out-of-9: F системи у стані відмови зображено на рис. 2.9.

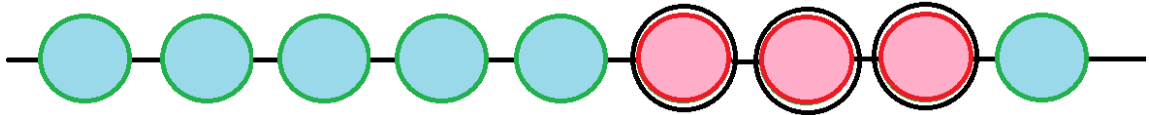


Рис. 2.9. 3-out-of-9: F система у стані відмови

Прикладом послідовної k -out-of- n : G системи є міст з n кабелями, де мінімум k кабелів, розташованих послідовно, забезпечують справний стан мосту. Приклад 3-out-of-9: F системи в роботоздатному стані зображено на рис. 2.10.

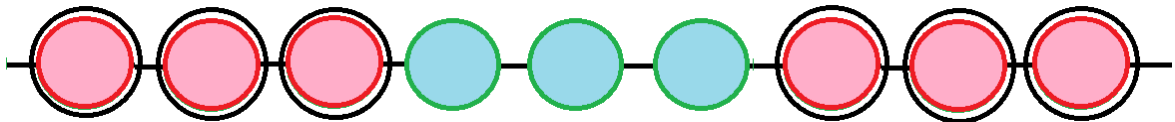


Рис. 2.10. 3-out-of-9: F система в роботоздатному стані

Для прикладу наведемо формули для розрахунку показників надійності k -out-of- n : G систем. Для таких систем релевантним є значення ймовірності безвідмовної роботи k її елементів, що можна розрахувати за наведеною формулою 2.11.

$$P(t, k, n) = \binom{n}{k} p(t)^k q(t)^{n-k} \quad (2.11)$$

, де n – кількість елементів в системі, $p(t)$ – ЙБР модуля системи, $q(t)$ може бути розраховано з виразу $1 - p(t)$, а $\binom{n}{k}$ – біноміальний коефіцієнт, що розраховують як $\frac{n!}{k!(n-k)!}$.

Таким чином ймовірність безвідмовної роботи не менше k елементів розраховують за формулою 2.12.

$$P(t, k, n) = \sum_{i=k}^n \binom{n}{i} p(t)^i q(t)^{n-i} \quad (2.12)$$

2.2. GL-моделі

Описані вище k -out-of- n системи не передбачають роботу з небазовими системами. Проте в реальному житті кількість модулів, за відмови (в стані роботоздатності) яких система відмовляє (зберігає роботоздатність) не є постійною.

Тому важливим завданням на теперішній час є створення таких моделей, що дозволяють описувати поведінку небазових систем в потоці відмов, та при цьому можуть бути створеними за допомогою деякого універсального алгоритму.

Одним з найбільш перспективних рішень даного завдання є створення так званих графо-логічних моделей (GL-моделей, які детально розглянуто в працях [1, 11-12]).

GL-моделлю відмовостійкої багатопроцесорної системи називають неорієнтований зв'язний граф G , кожному ребру якого відповідає булева функція, значення якої залежить від стану елементів системи.

Змінні для відображення роботоздатності елементів системи також називають індикаторними змінними x_i ($i = 1, \dots, n$). Їх значення визначаються за наступним правилом 2.13.

$$x_i = \begin{cases} 1, \text{ якщо елемент } x_i \text{ роботоздатний} \\ 0, \text{ якщо елемент } x_i \text{ відмовив} \end{cases} \quad (2.13)$$

Вважається, що ребро видаляється з графу у випадку, коли відповідна реберна функція набуває значення 0. В такій моделі індикатором роботоздатності системи є зв'язність графу.

Ці моделі дозволяють скоротити час на виконання статистичних експериментів через спрощення структури графу у порівнянні з структурою реальної системи, тому що граф не обов'язково повинен відповідати схемі модулів в реальній ВБС.

Загальним завданням в процесі створення GL-моделей є знаходження такого графу, щоб втрата зв'язності цим графом відбувалася тоді і тільки тоді, коли з'являється вектор стану, за якого реальна ВБС відмовляє. Для побудови GL-моделей систем повинні бути вирішені наступні питання:

- створення структури графу;
- вибір реберних функцій.

Для створення оптимальної структури графу необхідно зрозуміти, що GL-модель може бути побудована на основі будь-якого зв'язного графу, що складається з мінімум одного ребра. Таким чином можлива побудова багатьох різноманітних GL-моделей для однієї конкретної ВБС.

Найпростішим прикладом GL-моделі для 0-відмовостійкої системи є граф у вигляді ланцюга, реберні функції якого мають вигляд $f_i = x_i$. Втрата зв'язності графом відбувається при відмові будь-якого елементу системи. Приклад такої моделі зображено на рис. 2.11.

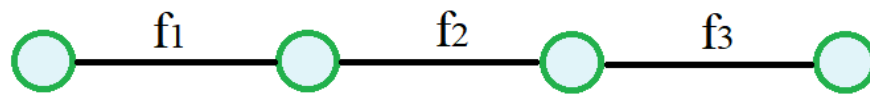


Рис. 2.11. GL-модель для 0-відмовостійкої системи, що складається з 3 елементів

Широке застосування отримали так звані циклічні моделі, що мають форму циклічного графу (кола). Реберні функції можуть мати будь-який вигляд за потреби. Для прикладу на рис. 2.12. зображено GL-модель для 1-відмовостійкої системи, що складається з 4 елементів. Реберні функції в такому випадку повинні мати вигляд $f_i = x_i$.

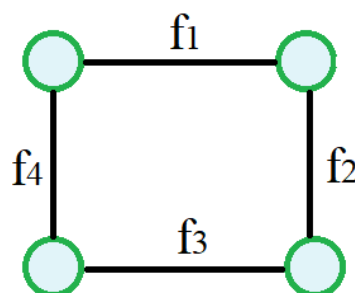


Рис. 2.12. GL-модель для 1-відмовостійкої системи, що складається з 4 елементів.

Для 2- та більше відмовостійких ВБС структура графу моделі може бути значно складнішою. Для прикладу на рис. 2.13.-2.15. зображені відповідно: GL-моделі на основі графу у вигляді колесу, у вигляді тримірного кубу та на основі подвійного циклу. Інший приклад моделі на основі графу типу подвійного циклу зображено на рис 2.16.

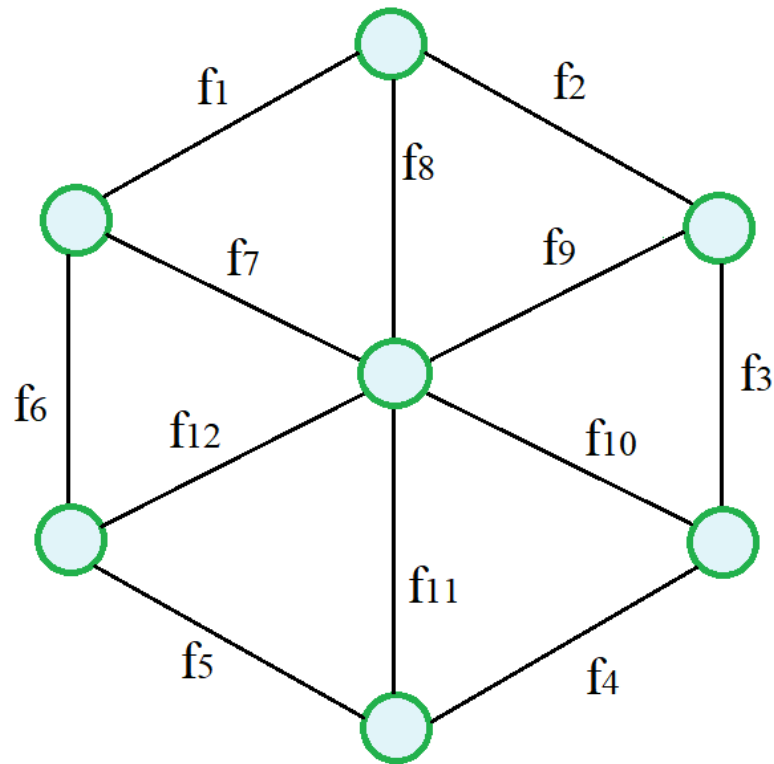


Рис. 2.13. Модель на основі графу у вигляді колесу

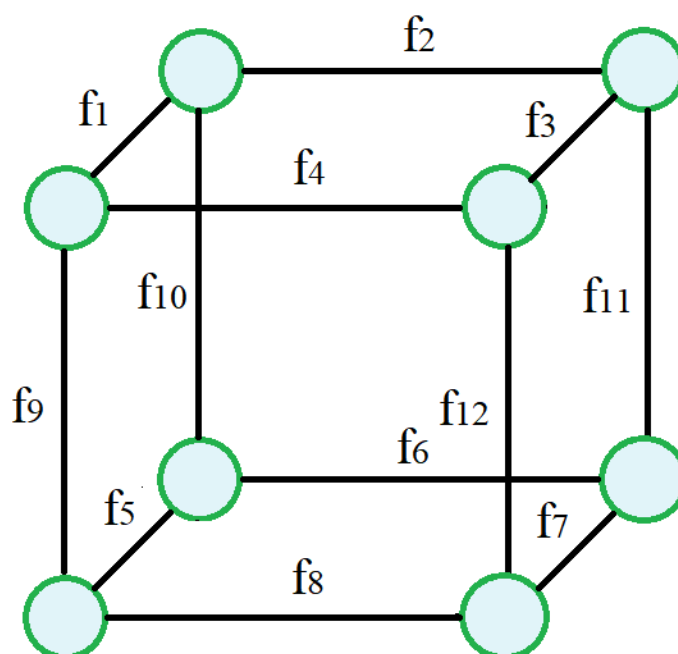


Рис. 2.14. Модель на основі графу у вигляді тримірного кубу

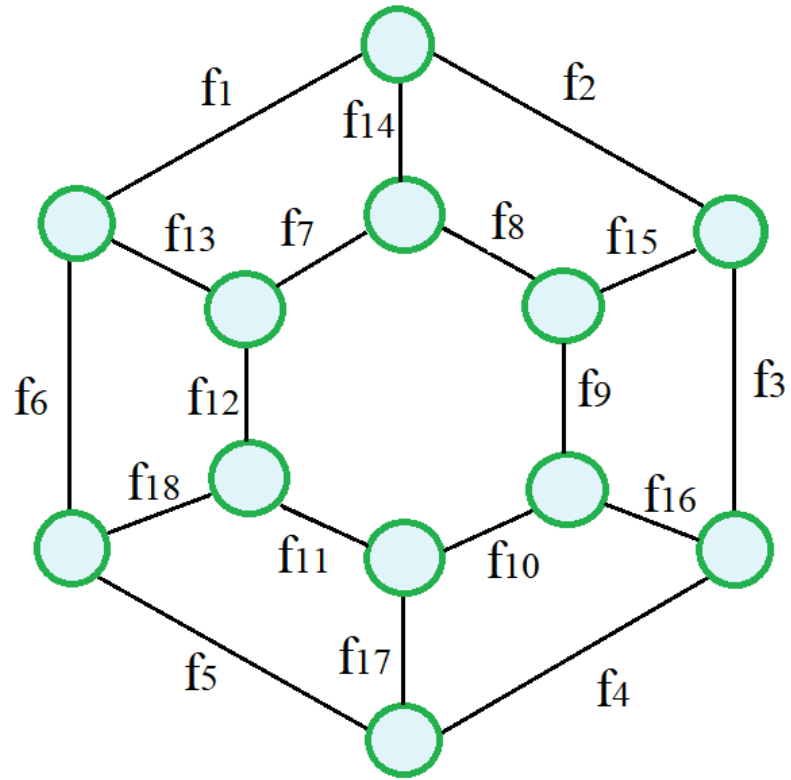


Рис. 2.15. Модель на основі графу у вигляді подвійного циклу

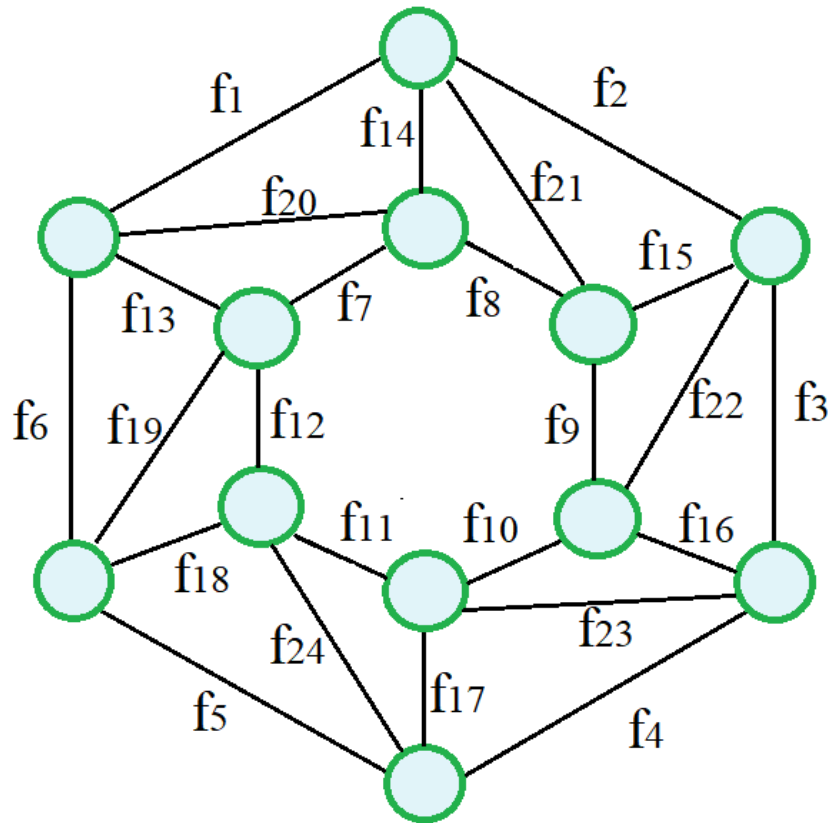


Рис. 2.16. Інший приклад моделі на основі графу у вигляді подвійного циклу

2.3. Побудова GL-моделей базових ВБС

Розглянемо спосіб формування GL-моделей для 2-відмовостійких багатопроцесорних систем. В якості графу оберемо циклічний граф з n ребрами, де n – кількість модулів в системі.

Було доведено, що в такому випадку для коректного відображення поведінки системи в потоці відмов реберні функції повинні мати наступний вигляд:

$$f_i(x_1, \dots, x_n) = \psi_i \vee \varphi_i \quad (2.14)$$

В формулі 2.14 ψ_i та φ_i можуть бути отриманими з виразів 2.15.

$$\psi_i = \bigvee_{j=1}^{\lfloor \frac{n-1}{2} \rfloor} x_{(i+j) \bmod(n)} \quad (2.15)$$

$$\varphi_i = x_i$$

В наведених вище формулах $i = (\overline{1, n})$.

За формулами отримаємо модель $K(2,8)$ системи, граф якої зображено на рис. 2.17.

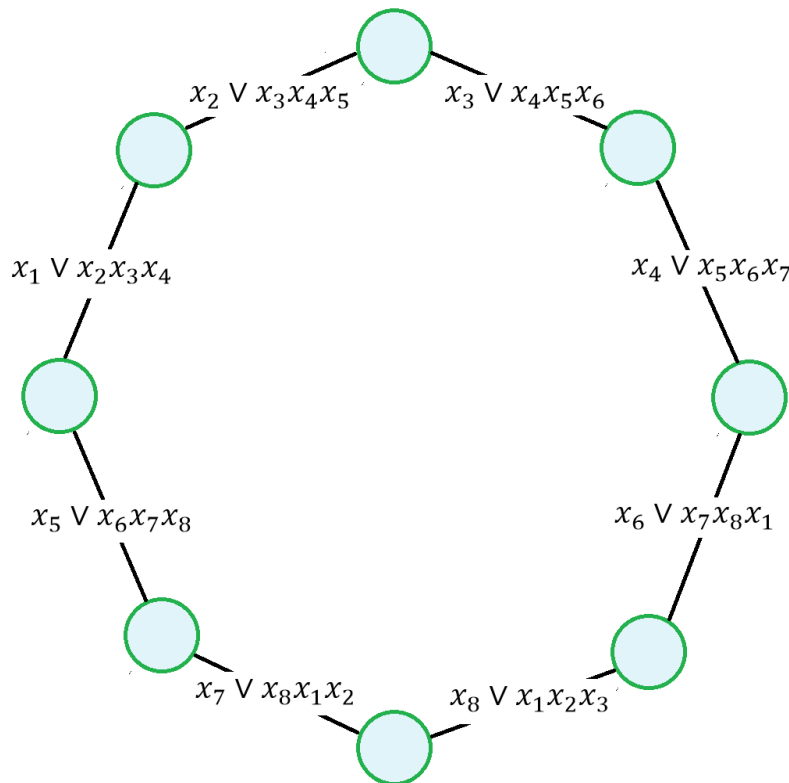


Рис. 2.17. Приклад GL-моделі 2-відмовостійкої ВБС, що містить в собі 8 модулів

Реберні функції для такого циклічного графу будуть мати наступний вигляд:

$$\begin{aligned}
 f_1 &= x_1 \vee x_2 x_3 x_4 \\
 f_2 &= x_2 \vee x_3 x_4 x_5 \\
 f_3 &= x_3 \vee x_4 x_5 x_6 \\
 f_4 &= x_4 \vee x_5 x_6 x_7 \\
 f_5 &= x_5 \vee x_6 x_7 x_8 \\
 f_6 &= x_6 \vee x_7 x_8 x_1 \\
 f_7 &= x_7 \vee x_8 x_1 x_2 \\
 f_8 &= x_8 \vee x_1 x_2 x_3
 \end{aligned}
 \tag{2.16}$$

Кількість отриманих реберних функцій відповідає кількості модулів в системі.

Якщо уважно придивитись до реберних функцій 2.16, то можна помітити, що кожен наступну функцію f_{i+1} можна отримати шляхом додавання 1 та виконання операції $\text{mod } n$ після над кожною змінною попередньої реберної функції f_i .

Для випадку 2-відмовостійких систем такий спосіб проектування моделей є простим та дозволяє отримати неповторні функції, що у випадку, коли $m > 2$, на жаль, не є справедливим. Окрім того, цей спосіб не є універсальним для будь-яких значень n та m , хоча все ж таки деякі рішення для $m = 3$ (2.17) та $m = 4$ (2.18) і були сформовані.

$$\begin{aligned}
 f_1 &= x_1 \vee x_2 \vee x_3 x_4, n = 6 \\
 f_1 &= x_1 \vee x_2 x_3 \vee x_5 x_7, n = 7
 \end{aligned}
 \tag{2.17}$$

$$\begin{aligned}
 f_1 &= x_1 \vee x_2 \vee x_3 \vee x_4 x_5, n = 6 \\
 f_1 &= x_1 \vee x_2 \vee x_3 x_4 \vee x_5 x_6, n = 7 \\
 f_1 &= x_1 \vee x_2 x_3 x_4 x_5 (x_6 \vee x_7) \vee (x_8 \vee \overline{x_4} \vee \overline{x_5}) (x_9 \vee \\
 &\quad \vee \overline{x_5}) x_{14} x_{10} x_{11} x_{12} x_{13}, n = 14
 \end{aligned}
 \tag{2.18}$$

В 2.17. та 2.18. наведені лише реберні функції для першого ребра тому, що описане правило перетворення наступних функцій f_{i+1} з f_i є справедливим і для цих випадків.

Через описані вище обмеження для випадків, коли це є необхідним, треба використовувати інші алгоритми для формування GL-моделей, наприклад рекурсивний алгоритм послідовного поділу на групи, який є універсальним для систем будь-якого ступеня відмовостійкості. Вказаний алгоритм більш детально буде розглянуто в наступному розділі.

Висновки розділу 2

В даному розділі проаналізовані різні структури моделей ВБС. Встановлено, що найбільш простими моделями є моделі, засновані на паралельно-послідовних структурах, які проте часто не можуть бути використаними для відображення поведінки реальної системи через те, що структура багатопроцесорних систем в наш час є значно складнішою та містить дуже велику кількість модулів.

Розглянуті GL-моделі, які є конкурентною альтернативою моделям із простою структурою, в свою чергу дозволяють відображати поведінку так званих небазових систем, якими в наш час є більшість ВБС. Крім того, такі моделі можуть мати значно простішу структуру, ніж структура реальної системи за рахунок «перенесення» складності структури на реберні функції. Вказаний факт дозволяє провести більшу кількість експериментів над моделями за фіксовану кількість часу та збільшити точність розрахунку показників надійності системи відповідно.

В даному розділі також було досліджено спосіб побудови GL-моделей базових систем для 2-відмовостіких систем та встановлено, що даний спосіб не може бути використаним для будь-яких значень m та n .

3. РОЗРОБЛЕНИЙ МЕТОД ПЕРЕТВОРЕННЯ GL-МОДЕЛЕЙ

3.1. Алгоритм послідовного поділу елементів системи на групи для формування GL-моделей

Алгоритм послідовного поділу на групи [1] є універсальним для будь-яких значень k , рекурсивним алгоритмом для формування GL-моделей k -ВБС. В якості графу використовується граф циклічного типу, для якого рекурсивним поділом елементів системи на групи конструюють реберні функції.

Для формування GL-моделей за даним алгоритмом необхідно виконати наступні кроки:

1. Спочатку розіб'ємо усі n елементів системи k -ВБС на довільну кількість підмножин t , що не мають перетинатися, та позначимо ці множини: n_1, n_2, \dots, n_t . Кожна з них містить відповідно $|n_1|, |n_2|, \dots, |n_t|$. Таким чином $\sum_{i=1}^t n_i = n$.
2. Розглянемо усі можливі комбінації значень m_1, m_2, \dots, m_t , що відповідають розподілу відмов між підмножинами. Загальна кількість відмов в групах повинна відповідати початковій відмовостійкості системи (3.1)

$$\sum_{i=1}^t m_i = m, m_i \leq n_i \quad (3.1)$$

3. Для кожного варіанту розподілу необхідно записати вираз вигляду 3.2.

$$\begin{aligned} K(m_1, n_1) \times K(m_2, n_2) \times \dots \times K(m_t, n_t) = \\ = \{(x_1 \times y_1 \times \dots \times z_1), (x_1 \times y_1 \times \dots \times z_2) \dots \\ \dots (x_{n_1} \times y_{n_2} \times \dots \times z_{n_t})\} \end{aligned} \quad (3.2)$$

У виразі 3.2 для $x_i, y_j \dots z_k$ справедливими є:

$$i = \overline{1, C_{n_1}^{m_1}}, j = \overline{1, C_{n_2}^{m_2}} \dots k = \overline{1, C_{n_t}^{m_t}} \quad (3.3)$$

4. Перетворити вирази, отримані в попередньому пункті в реберні функції за наступним правилом: якщо $K_{|n_j|}^{m_{ij}}$ відображає поведінку m_{ij} -відмовостійкої багатопроцесорної системи, що складається з $|n_j|$

елементів, то система результуючих функцій набуватиме вигляду, що наведено в 3.4.

[illegible]

Рекурсивний поділ елементів груп на підгрупи та розподіл між ними відмов відбувається доти, доки $m_{ij} = 1$, або $m_{ij} = n_j$. За таких умов можна виконати еквівалентні перетворення 3.5-3.7.

$$K_{|n_j|}^0 = 0 \quad (3.5)$$

$$K_{|n_j|}^1 = \bigwedge_{k \in n_j} x_k \quad (3.6)$$

$$K_{|n_j|}^{|m_j|} = \bigvee_{k \in n_i} x_k \quad (3.7)$$

Програмно-орієнтований алгоритм послідовного поділу елементів системи на групи зображено на рис. 3.1.

На базі зазначеного алгоритму розглянемо побудову реберних функцій кільцевого графу для 3-відмовостійкої системи, що містить 8 елементів. Для прикладу розіб'ємо елементи системи на підмножини $\{1, 2, 3\}$ та $\{4, 5, 6, 7, 8\}$. Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для цих груп наведені в таблиці 3.1.

Таким чином для $K(3, 8)$ отримаємо:

$$\begin{cases} x_1 \vee x_2 \vee x_3 \\ K(2, 3) \times (x_4 x_5 x_6 x_7 x_8) \\ x_1 x_2 x_3 \times K(2, 5) \\ K(3, 5) \end{cases} \quad (3.8)$$

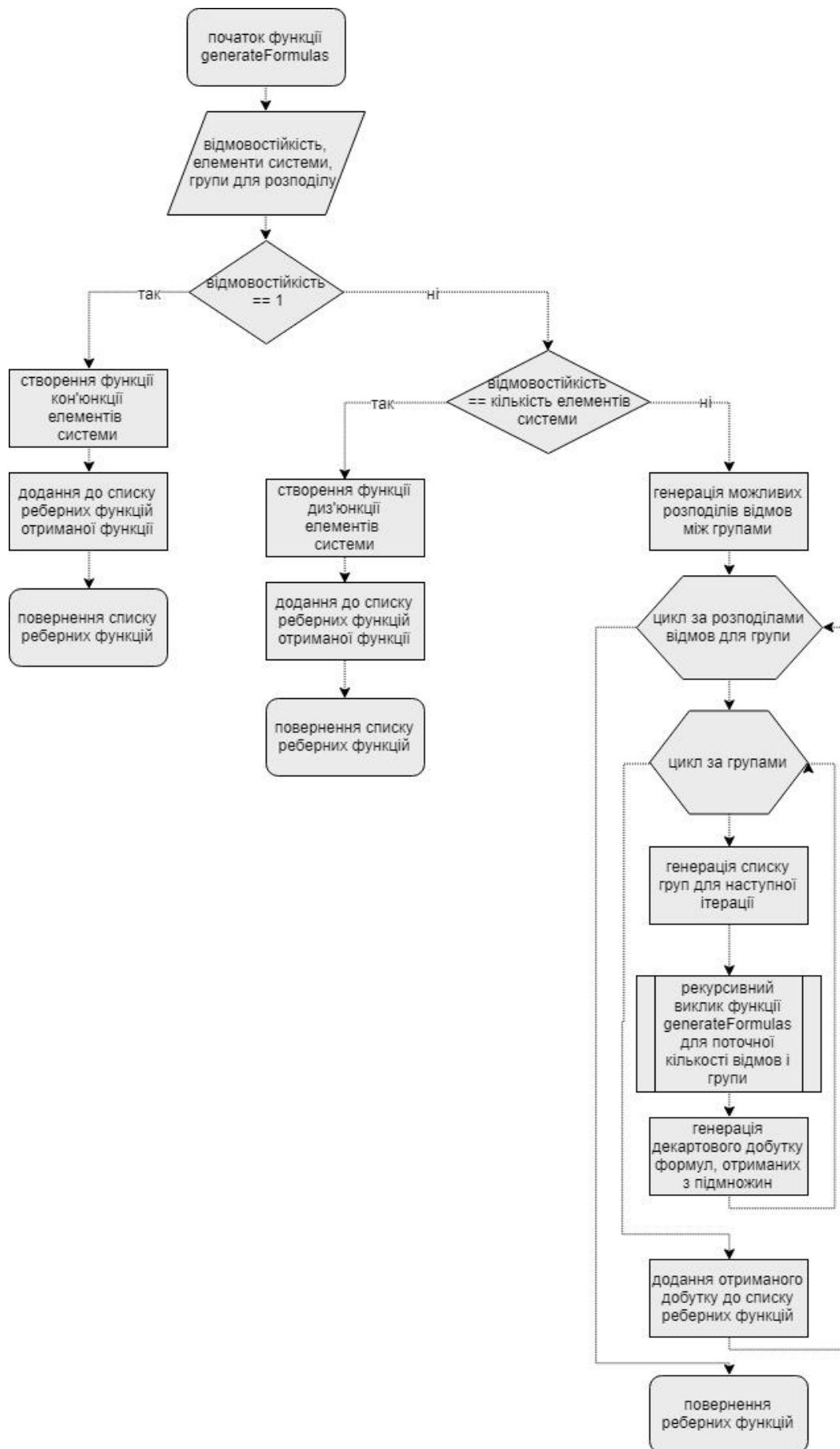


Рис. 3.1. Програмно-орієнтований алгоритм послідовного поділу елементів системи на групи

Таблиця 3.1

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов на підмножинах $\{1, 2, 3\}$ та $\{4, 5, 6, 7, 8\}$

$n_1 = \{1, 2, 3\}$	$n_2 = \{4, 5, 6, 7, 8\}$
m_1	m_2
$K(3,3)$	
$K(2,3)$	$K(1,5)$
$K(1,3)$	$K(2,5)$
	$K(3,5)$

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для поділу на підмножини $\{1, 2\}$ та $\{3\}$ підсистеми $K(2,3)$ продемонстровані в таблиці 3.2.

Таблиця 3.2

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для підмножин $\{1, 2\}$ та $\{3\}$ $K(2,3)$

$n_1 = \{1, 2\}$	$n_2 = \{3\}$
m_1	m_2
$K(2,2)$	
$K(1,2)$	$K(1,1)$

Таким чином отримана система реберних функцій для підсистеми $K(2,3)$ буде мати вигляд 3.9.

$$\begin{cases} x_1 \vee x_2 \\ x_1 x_2 \vee x_3 \end{cases} \quad (3.9)$$

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для поділу підмножини $K(2,5)$ наведені в таблиці 3.3.

Таблиця 3.3

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для підмножин $\{4, 5\}$ та $\{6, 7, 8\}$ $K(2,5)$

$n_1 = \{4, 5\}$	$n_2 = \{6, 7, 8\}$
m_1	m_2
$K(2,2)$	
$K(1,2)$	$K(1,3)$
	$K(2,3)$

Таким чином для множини $K(2,5)$ отримаємо вирази 3.10.

$$\begin{cases} x_4 \vee x_5 \\ x_4 x_5 \vee x_6 x_7 x_8 \\ K(2,3) \end{cases} \quad (3.10)$$

Розіб'ємо підмножину $\{6, 7, 8\}$ на підмножини $\{6, 7\}$ та $\{8\}$. Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для поділу підмножини $K(2,3)$ на вказані підмножини наведені в таблиці 3.4.

Таблиця 3.4

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для підмножин $\{6, 7\}$ та $\{8\}$ $K(2,3)$

$n_1 = \{6, 7\}$	$n_2 = \{8\}$
m_1	m_2
$K(2,2)$	
$K(1,2)$	$K(1,1)$

З таблиці 3.4. отримаємо 3.11.

$$\begin{cases} x_6 \vee x_7 \\ x_6 x_7 \vee x_8 \end{cases} \quad (3.11)$$

Розіб'ємо підмножину $\{4, 5, 6, 7, 8\}$ на підмножини $\{4, 5\}$ та $\{6, 7, 8\}$. Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для

поділу підмножини $K(3,5)$ на підмножини $\{4, 5\}$ та $\{6, 7, 8\}$ наведені в таблиці 3.5.

Таблиця 3.5

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для підмножин $\{4, 5\}$ та $\{6, 7, 8\}K(3,5)$

$n_1 = \{4, 5\}$	$n_2 = \{6, 7, 8\}$
m_1	m_2
$K(2,2)$	$K(1,3)$
$K(1,2)$	$K(2,3)$
	$K(3,3)$

Таким чином для множини $K(2,3)$ отримаємо:

$$\begin{cases} x_4 \vee x_5 \vee x_6 x_7 x_8 \\ x_4 x_5 \vee K(2, 3) \\ x_6 \vee x_7 \vee x_8 \end{cases} \quad (3.12)$$

Тепер умовам виходу з рекурсії на відповідає тільки $K(2,3)$ для підмножини $\{6, 7, 8\}$. Розіб'ємо підмножину $\{6, 7, 8\}$ на підмножини $\{6, 7\}$ та $\{8\}$.

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для поділу підмножини $K(2,3)$ на вказані підмножини наведені в таблиці 3.4.

Результуючі реберні функції можна отримати з наведених вище таблиць 3.1 – 3.5:

$$\begin{aligned} f_1 &= x_1 \vee x_2 \vee x_3 \\ f_2 &= x_1 \vee x_2 \vee x_4 x_5 x_6 x_7 x_8 \\ f_3 &= x_1 x_2 \vee x_3 \vee x_4 x_5 x_6 x_7 x_8 \\ f_4 &= x_4 \vee x_5 \vee x_1 x_2 x_3 \\ f_5 &= x_6 \vee x_7 \vee x_1 x_2 x_3 \\ f_6 &= x_4 x_5 \vee x_6 x_7 x_8 \vee x_1 x_2 x_3 \\ f_7 &= x_6 x_7 \vee x_8 \vee x_1 x_2 x_3 \end{aligned} \quad (3.13)$$

$$f_8 = x_4 \vee x_5 \vee x_6 x_7 x_8$$

$$f_9 = x_6 \vee x_7 \vee x_4 x_5$$

$$f_{10} = x_6 x_7 \vee x_8 \vee x_4 x_5$$

$$f_{11} = x_6 \vee x_7 \vee x_8$$

Очевидно, що кількість отриманих за алгоритмом послідовного поділу на групи реберних функцій більше кількості елементів реальної ВБС, проте функції є коротшими та безповторними.

Але, незважаючи на простоту формування реберних функцій за цим алгоритмом, необхідною є методика визначення підмножин, на які потрібно ділити елементи в кожній ітерації. Для реальних ВБС з великою кількістю модулів варіантів розбиття елементів системи на підмножини може бути дуже багато.

Для прикладу наведемо отримання реберних функцій для 2-відмовостійкої системи, що складається з 6 елементів, за умови початкового розбиття елементів системи на рівні за потужністю підмножини. В нашому випадку розіб'ємо елементи системи на в першій ітерації алгоритму підмножини $\{1, 2, 3\}$ та $\{4, 5, 6\}$.

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для початкового розбиття на рівні за потужністю підмножини наведені в таблиці 3.6.

Таблиця 3.6

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для підмножин $\{1, 2, 3\}$ та $\{5, 6, 7\}$

$n_1 = \{1, 2, 3\}$	$n_2 = \{4, 5, 6\}$
m_1	m_2
$K(2,3)$	
$K(1,3)$	$K(1,3)$
	$K(2,3)$

Таким чином для множини $K(2, 6)$ отримаємо 3.14.

$$\begin{cases} K(2, 3) \\ x_1 x_2 x_3 \times x_4 x_5 x_6 \\ K(2, 3) \end{cases} \quad (3.14)$$

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для поділу на підмножини $\{1, 2\}$ та $\{3\}$ $K(2,3)$ наведені в таблиці 3.7. Таким чином для розбиття елементів на дані підмножини отримаємо 3.15.

$$\begin{cases} x_1 \vee x_2 \\ x_1 x_2 \vee x_3 \end{cases} \quad (3.15)$$

Розіб'ємо підмножину $\{4, 5, 6\}$ на підмножини $\{4, 5\}$ та $\{6\}$. Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для поділу підмножини $K(2,3)$ на вказані підмножини наведені в таблиці 3.7.

Таблиця 3.7

Комбінації $K_{|n_j|}^{m_{ij}}$, що відповідають різним варіантам розподілення відмов для підмножин $\{4, 5\}$ та $\{6\}$ $K(2,3)$

$n_1 = \{4, 5\}$	$n_2 = \{6\}$
m_1	m_2
$K(2,2)$	
$K(1,2)$	$K(1,1)$

З таблиці 3.7. отримаємо:

$$\begin{cases} x_4 \vee x_5 \\ x_4 x_5 \vee x_6 \end{cases} \quad (3.16)$$

Результуючі мають вигляд 3.17.

$$\begin{aligned} f_1 &= x_1 \vee x_2 \\ f_2 &= x_1 x_2 \vee x_3 \\ f_3 &= x_1 x_2 x_3 \vee x_4 x_5 x_6 \\ f_4 &= x_4 \vee x_5 \\ f_5 &= x_4 x_5 \vee x_6 \end{aligned} \quad (3.17)$$

Як зазначається автором роботи [13], саме такий варіант розбиття модулів на підмножини (на рівні, або майже рівні за потужністю групи) дозволяє отримати найпростіші моделі.

3.2. Побудова GL-моделей для небазових ВБС

Алгоритми, що були розглянуті раніше, можуть бути використаними для побудови моделей базових систем. Так, алгоритм послідовного поділу на групи використовує комбінації розподілу відмов за групами і оперує фіксованою кількістю відмов незалежно від елементів, що знаходяться в цих групах. Тому з його допомогою неможливо, наприклад, побудувати модель системи, що стійка до появи 3 відмов та деяких векторів з 4 відмовами. Реальні ж ВБС часто є небазовими. Саме тому дослідження можливості модифікації існуючих алгоритмів для вирішення завдання побудови моделей небазових систем є актуальними в наш час.

Для цього в загальному випадку необхідно підібрати базову ВБС, що є найбільш близькою до реальної та модифікувати її модель або за допомогою модифікації графу (проведення додаткових ребер з відповідними булевими функціями у графі) або ж за допомогою модифікації реберних функцій.

У випадку, коли існує декілька векторів, на яких система поводить себе як небазова, модифікацію моделі можна виконувати крок за кроком з наступною мінімізацією реберних функцій [14-16], якщо це можливо.

Відображення підвищення ступені відмовостійкості відносно базової ВБС може бути виконано за допомогою модифікації реберних функцій. Якщо існуюча m -відмовостійка система зберігає роботоздатність за появи деякого вектора, що містить $m + 1$ нулів (в якому $x_1 \dots x_{m+1}$ набувають значення 0), тоді для збереження зв'язності графом першим кроком необхідно знайти та модифікувати деякі з функцій, що набувають значення 0 за допомогою додавання константи 3.18.

$$f'_i = f_i \vee (\overline{x_1} \wedge \overline{x_2} \wedge \dots \wedge \overline{x_{m+1}} \wedge x_{m+2} \wedge \dots \wedge x_n) \quad (3.18)$$

Для циклічного графу, де втрата зв'язності виникає за умови втрати 2 ребер, необхідно модифікувати всі функції, що набувають значення 0 крім будь-якої одної.

Програмно-орієнтований алгоритм для відображення підвищення ступеня відмовостійкості реберних функцій за допомогою модифікації реберних функцій зображено на рис. 3.2.

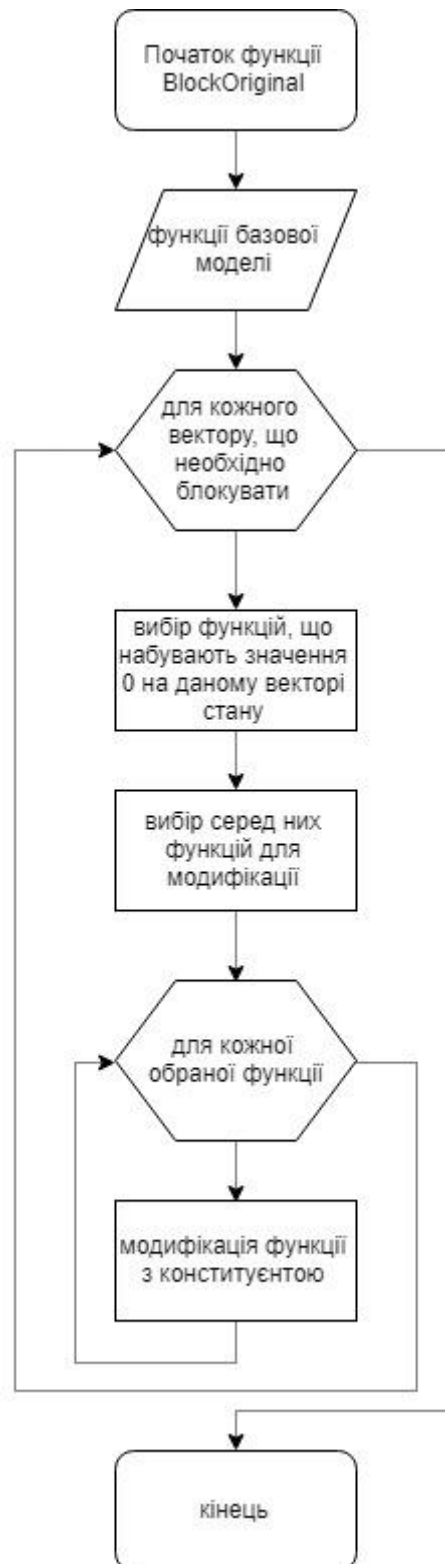


Рис. 3.2. Програмно-орієнтований алгоритм для відображення підвищення ступеня відмовостійкості реберних функцій за допомогою модифікації реберних функцій

На прикладі базової моделі $K(3,8)$, система реберних функцій якої наведена в 3.13, розглянемо систему, яка також повинна зберігати роботоздатний стан за появи вектора 3.19.

$$\begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \quad (3.19)$$

Тоді конститuenta, яку необхідно додати до функцій, що набувають значення 0, буде мати вигляд 3.20.

$$\overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} x_5 x_6 x_7 x_8 \quad (3.20)$$

Таким чином необхідно модифікувати будь-які 2 з 3 функцій f_1 , f_2 або f_3 . Тоді реберні функції небазової моделі можуть мати вигляд 3.21.

$$\begin{aligned} f_1 &= x_1 \vee x_2 \vee x_3 \vee \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} x_5 x_6 x_7 x_8 \\ f_2 &= x_1 \vee x_2 \vee x_4 x_5 x_6 x_7 x_8 \vee \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} x_5 x_6 x_7 x_8 \\ f_3 &= x_1 x_2 \vee x_3 \vee x_4 x_5 x_6 x_7 x_8 \\ f_4 &= x_4 \vee x_5 \vee x_1 x_2 x_3 \\ f_5 &= x_6 \vee x_7 \vee x_1 x_2 x_3 \\ f_6 &= x_4 x_5 \vee x_6 x_7 x_8 \vee x_1 x_2 x_3 \\ f_7 &= x_6 x_7 \vee x_8 \vee x_1 x_2 x_3 \\ f_8 &= x_4 \vee x_5 \vee x_6 x_7 x_8 \\ f_9 &= x_6 \vee x_7 \vee x_4 x_5 \\ f_{10} &= x_6 x_7 \vee x_8 \vee x_4 x_5 \\ f_{11} &= x_6 \vee x_7 \vee x_8 \end{aligned} \quad (3.21)$$

Відображення підвищення ступеня відмовостійкості системи іншим способом (за допомогою проведення внутрішніх ребер у графі) виконується за наступним алгоритмом:

1. Визначення реберних функцій, що набувають значення 0.
2. Визначення необхідної кількості ребер, які потрібно додати до графу таким чином, щоб між двома найближчим вузлами, що інцидентні двом довільним додатковим ребрам, не знаходилось 2 або більше ребра, що набувають значення 0.

3. Надання у відповідність знайденим в пункті 2 додатковим ребрам функції конституенти одиниці (3.22) для вектора стану, що необхідно блокувати.

$$f'_i = f_i \vee (\overline{x_1} \wedge \overline{x_2} \wedge \dots \wedge \overline{x_{m+1}} \wedge x_{m+2} \wedge \dots \wedge x_n) \quad (3.22)$$

Введені додаткові ребра не впливають на поведінку моделі за появи інших векторів, проте забезпечують шлях між ділянками кільця, які відокремлені через появу вектора, що розглядається.

Для прикладу розглянемо модель базової $K(2, 6)$ системи, яку зображено на рис 3.3.

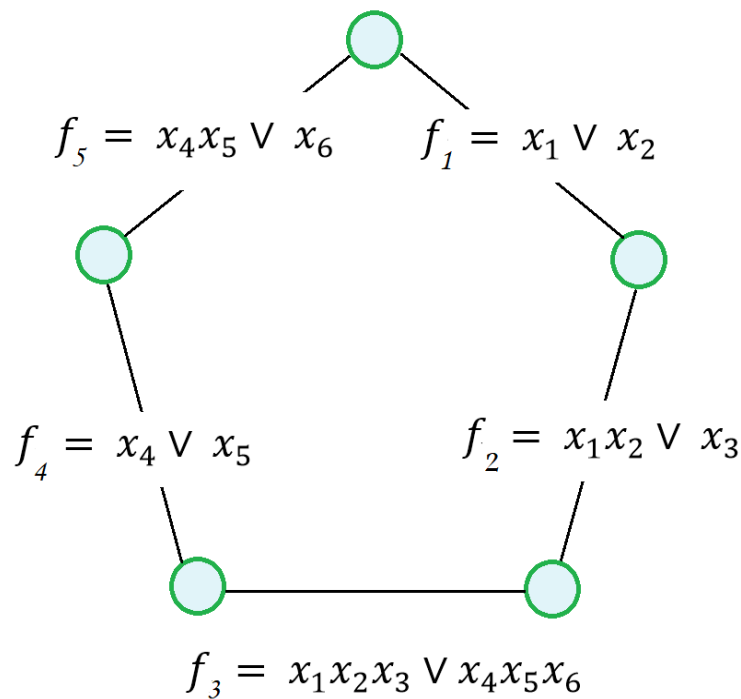


Рис. 3.3. Приклад GL-моделі 2-відмовостійкої ВБС, що складається з 6 елементів

Нехай нам необхідно модифікувати цю модель таким чином, щоб вона відповідала $K(2, 6)$ системі, яка зберігає роботоздатний стан за умови появи вектора 3.23.

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{array} \quad (3.23)$$

Також вона залишається роботоздатною за появи вектора 3.24.

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{array} \quad (3.24)$$

Тоді за появи вектора 3.23 функції f_1 та f_2 набувають значення 0, а за появи вектора 3.24 – функції f_1 та f_3 . В такому випадку можливе, наприклад, проведення 2 додаткових ребер з функціями 3.25 та 3.26.

$$\overline{x_1} \overline{x_2} \overline{x_3} x_4 x_5 x_6 \quad (3.25)$$

$$\overline{x_1} \overline{x_2} x_3 \overline{x_4} x_5 x_6 \quad (3.26)$$

Таким чином після запропонованих перетворень модель системи набуває вигляду, як зображено на рис 3.4.

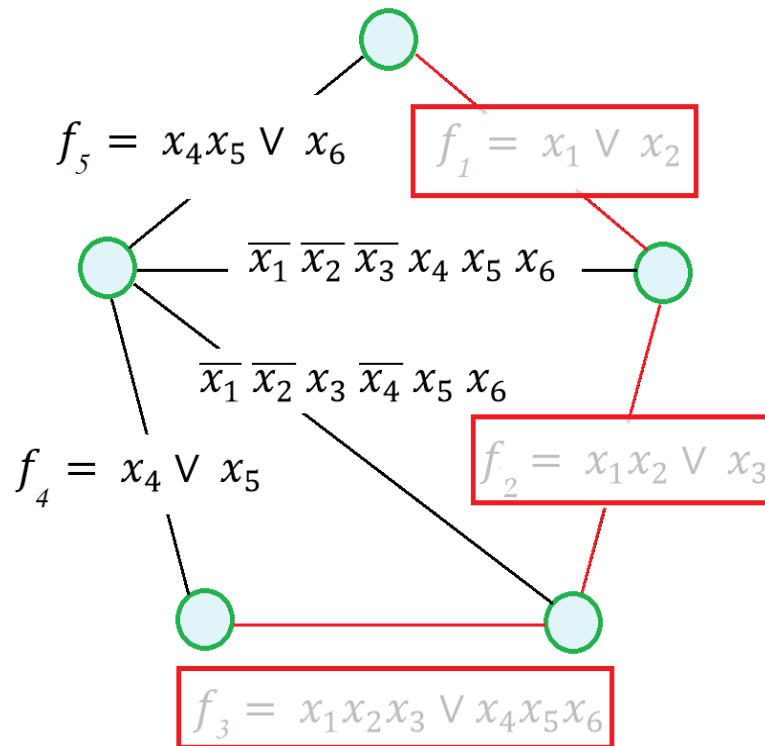


Рис. 3.4. Приклад GL-моделі 2-відмовостійкої ВБС, що складається з 6 елементів після модифікації.

Проблему відображення зниження ступеня відмовостійкості системи на моделі також можна вирішити або за допомогою модифікації реберних функцій, або за допомогою проведення додаткових ребер.

Завданням є побудова моделі для $K(m, n)$ системи, яка проте втрачає роботоздатність за появи деякого вектора (деяких векторів) з відмовами m модулів.

Тоді алгоритм відображення зниження ступеня відмовостійкості системи за допомогою модифікації реберних функцій буде складатися з наступних етапів:

1. Визначення реберних функцій, що набувають значення 1 (таких ребер повинно бути $n - 1$, де n – кількість ребер в базовій моделі).
2. Вибір будь-якої функції з множини, визначеної в пункті 1.
3. Модифікація даної функції за правилом 3.27.

$$f'_i = f_i(x_1 \vee x_2 \vee \dots \vee x_m) \quad (3.27)$$

На прикладі розглянутої вище $K(2,6)$ системи розглянемо завдання відображення зниження ступеня відмовостійкості, якщо реальна система відмовляє за появи вектора 3.28

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{array} \quad (3.28)$$

За появи вектора 3.28 функція f_1 набуває нульового значення, тому потрібно модифікувати будь-яку функцію з множини $\{f_2, f_3, f_4, f_5\}$. Тоді, наприклад, функція f_2 після модифікації буде мати вигляд 3.29, а після мінімізації вигляду 3.30.

$$f_2 = (x_1 x_2 \vee x_3)(x_1 x_2) \quad (3.29)$$

$$f_2 = x_1 x_2 \quad (3.30)$$

Задачу відображення зниження ступеня відмовостійкості вирішують також проведенням додаткового ребра в графі та додаванням ще однієї вершини, що з'єднана з іншою частиною графа доданим ребром.

Зазначеному ребру слід поставити у відповідність функцію, що дорівнює нулю на всіх наборах-векторах стану системи, за яких система відмовляє (наприклад, подану у вигляді КНФ). Таким чином на прикладі моделі $K(2,6)$ системи, яка, для прикладу, відмовляє за відмови елементів з номерами 1 та 2, зрозуміло, що для вирішення завдання відображення зниження ступеня відмовостійкості необхідно додати одну вершину та ребро з реберною функцією 3.31.

$$f_{add} = x_1 \vee x_2 \quad (3.31)$$

Тоді після перетворення системи в небазову за допомогою введення нового ребра та вершини GL-модель буде мати вигляд, як на рис. 3.5.

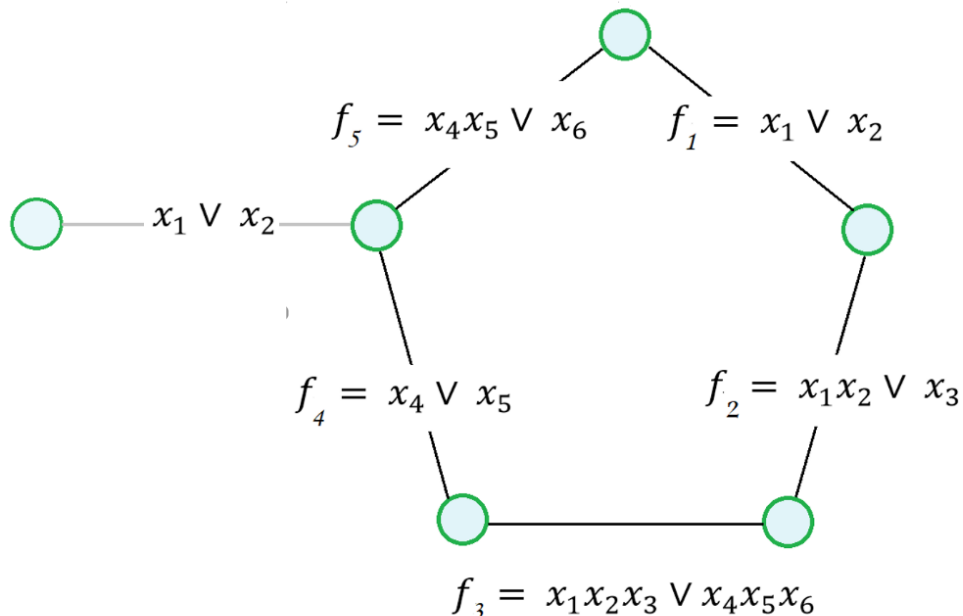


Рис. 3.5. Приклад GL-моделі 2-відмовостійкої ВБС, що складається з 6 елементів після перетворення її в небазову

3.3. Мінімізація GL-моделей

Як вже було зазначено раніше, спрощення GL-моделей дозволяє збільшити сумарну кількість експериментів над неї, що в свою чергу призводить до підвищення точності отриманих показників надійності системи, яку дана модель описує. Саме тому завдання мінімізації GL-моделей відмовостійких багатопроцесорних систем є дуже актуальним питанням.

В даному розділі розглянемо метод мінімізації GL-моделей, що заснований на операції склеювання та дозволяє зменшити не тільки кількість реберних функцій, але й загальну складність реберних функцій моделей.

Операція склеювання виконується за правилом 3.32

$$\left. \begin{array}{l} F_1 = A \vee \varphi_1 \\ F_2 = A \vee \varphi_2 \end{array} \right\} \Rightarrow F_3 = A \vee \varphi_1 \cdot \varphi_2 \quad (3.32)$$

У виразі 3.32 A , φ_1 , φ_2 – булеви вирази, з яких складаються реберні функції F_1 , F_2 , F_3 моделі.

Очевидно, що операцію склеювання можна виконати лише за наявності загальної частини A , що міститься у функціях для склеювання. Вираз A повинен бути поєднаним з рештою частини функції за допомогою операції «або».

Доведемо, що операція склеювання функцій F_1 та F_2 3.32 є справедливою і для випадку, коли F_1 та F_2 вже брали участь в склеюванні [17]:

1. Можна стверджувати, що F_3 дорівнює 0 тільки у випадку, якщо одна з функцій, що приймає участь в операції склеювання дорівнює 0. Справедливість цього твердження показана за допомогою таблиці істинності 3.8.

Таблиця 3.8

Таблиця істинності для функції F_3

A	φ_1	φ_2	F_1	F_2	F_3
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

2. Множини змінних, що знаходяться по різні сторони знаку диз'юнкції в функції F_3 не перетинаються. Це твердження є справедливим тому, що функції F_1 та F_2 канонічного представлення є неповторними, а склеювання виконується за виразом A , що не має загальних змінних з частинами φ_1 та φ_2 .
3. Розглянемо варіант появи вектора стану з m або менше нульовими компонентами. Кількість реберних функцій, що набувають значення 0 за появи таких векторів, за твердженням 2 не змінюється, а отже після мінімізації модель буде адекватно відображати поведінку системи за умови появи відмов елементів в кількості до m .
4. Розглянемо варіант, коли в системі відмовили більш, ніж m елементів. В такому випадку модель повинна відображати відмову системи в цілому. Існує два варіанту розвитку подій:

4.1. Якщо набуває значення 0 лише одна з функцій F_1 та F_2 , тоді справедливність вказаної вище операції склеювання очевидна через те, що за твердженням в пункті 1 F_3 також буде набувати значення 0.

4.2. Можна стверджувати, що за появи будь-якого вектора стану, що описує відмову $m + 1$ модулів системи і на якому функції F_1 та F_2 набувають значення 0 одночасно, існує ще хоча б одна реберна функція, яка також набуває значення 0. Нехай вектор V , за появи якого обидві функції набувають значення нуля, містить елементи $x_{v1}, x_{v2}, \dots, x_{vm}, x_{v(m+1)}$ рівні 0.

Нехай також вираз A набуває значення 0 за появи будь-якого вектора, де елементи $x_{v1}, x_{v2}, \dots, x_{vk}, (k < m)$ набувають значення 0, а вираз $\varphi_1 \cdot \varphi_2$ – за появи вектора з нулями на елементах $x_{v(k+1)}, \dots, x_{vm+1}$ (тобто на решті елементів, що набувають значення 0 у векторі V). Розглянемо інший вектор V' , що містить нулі на елементах $x_{v2}, x_{v3}, \dots, x_{vk}, x_{v(k+1)} \dots x_{v(m+1)}$. За появи такого вектора функції F_1 та F_2 набувають значення 1 через те, що $A \neq 0$.

У векторі V' m елементів набувають значення 0, тоді, як було згадано раніше, існує реберна функція F_x , що набуває значення 0 за появи такого вектора. Таким чином за появи вектора V буде існувати три функції F_1 , F_2 та F_x , що набувають значення 0 на векторі з $m + 1$ нулями.

Описана вище операція склеювання дозволяє спростити модель, проте варто зазначити, що в результаті можна отримати складнішу функцію, хоча і одну замість двох. Тому на даному етапі необхідно визначити критерії якості (оптимальності) моделей. Серед найбільш розповсюджених та практичних критеріїв якості моделі можна привести:

- мінімальна загальна складність моделі, що оцінюється сумарною складністю реберних функцій та числа ребер графу;

- складність комп'ютерного представлення моделі;
- мінімальна кількість ребер, яка видаляється з графу за появи векторів з m елементами, що набувають значення 0;
- мінімальна складність реберних функцій.

І хоча в даній дипломній роботі саме критерію мінімальної складності реберних функцій приділено увагу, необхідно також розуміти, яким чином можливо створити модель, що відповідає декільком критеріям оптимальності одночасно.

Для вирішення такої проблеми існує методика, що детально описана в [18]. Вона дозволяє виконати оптимізацію GL-моделей за прийнятний час з використанням таблиць, які дуже схожі на імплікативну матрицю Квайна.

Методика передбачає виконання наступної послідовності дій:

- виконання усіх можливих склеювань функцій канонічної моделі системи;
- створення двох таблиць, рядки яких повинні відповідати функціям базової моделі системи і функціям, отриманим після виконання всіх можливих склеювань;
- стовпчикам однієї таблиці повинні відповідати усі можливі вектори з m нулями, стовпчикам другої таблиці в свою чергу – усі можливі вектори з $m + 1$ нулями;
- для кожної таблиці необхідно розглянути кожний рядок і поставити відмітку, якщо дана функція набуває нульового значення за появи відповідного вектора стану;
- вибір сукупності функцій, що в першій таблиці має одну та менше позначок в кожному стовпчику таблиці, а в другій – дві або більше, та одночасно відповідає потрібному критерію або ж сукупності обраних критеріїв.

Дана методика забезпечує наявність лише однієї функції, що набуває значення 0 за появи вектора стану системи з m нулями, завдяки інформації з першої таблиці та наявність мінімум двох функцій, що набувають значення 0 за

вектора стану з $m + 1$ нулями завдяки другій таблиці. Таким чином сукупність результуючих реберних функцій повністю відповідає моделі базової $K(m, n)$ системи.

На прикладі моделі системи $K(3, 7)$, що містить за алгоритмом послідовного поділу на групи реберні функції 3.33, розглянемо функціонування розглянутої методики.

Початковий поділ елементів системи на групи виконується таким чином: $\{x_1, x_2, x_3\}$ та $\{x_4, x_5, x_6, x_7\}$.

$$\begin{aligned}
 f_1 &= x_1 \vee x_2 \vee x_3 \\
 f_2 &= x_1 \vee x_2 \vee x_4 x_5 x_6 x_7 \\
 f_3 &= x_1 x_2 \vee x_3 \vee x_4 x_5 x_6 x_7 \\
 f_4 &= x_4 \vee x_5 \vee x_1 x_2 x_3 \\
 f_5 &= x_6 \vee x_7 \vee x_1 x_2 x_3 \\
 f_6 &= x_4 x_5 \vee x_6 x_7 \vee x_1 x_2 x_3 \\
 f_7 &= x_4 \vee x_5 \vee x_6 x_7 \\
 f_8 &= x_6 \vee x_7 \vee x_4 x_5
 \end{aligned} \tag{3.33}$$

Після виконання усіх можливих склеювань функцій канонічної моделі системи було сформовано наведені в 3.34. булеві функції.

$$\begin{aligned}
 f_9 &= x_3 \vee (x_1 \vee x_2)(x_1 x_2 \vee x_4 x_5 x_6 x_7) \\
 f_{10} &= x_1 \vee x_2 \vee x_3 x_4 x_5 x_6 x_7 \\
 f_{11} &= x_4 x_5 x_6 x_7 \vee (x_1 \vee x_2)(x_1 x_2 \vee x_3) \\
 f_{12} &= x_1 x_2 x_3 \vee (x_4 \vee x_5)(x_6 \vee x_7)(x_4 x_5 \vee x_6 x_7) \\
 f_{13} &= x_4 \vee x_5 \vee x_1 x_2 x_3 x_6 x_7 \\
 f_{14} &= x_6 \vee x_7 \vee x_1 x_2 x_3 x_4 x_5 \\
 f_{15} &= x_6 x_7 \vee (x_1 x_2 x_3 \vee x_4 x_5)(x_4 \vee x_5) \\
 f_{16} &= x_4 x_5 \vee (x_1 x_2 x_3 \vee x_6 x_7)(x_6 \vee x_7) \\
 f_{17} &= x_1 x_2 x_3 \vee (x_4 \vee x_5)(x_6 \vee x_7)
 \end{aligned} \tag{3.34}$$

Сформуємо таблиці, вказані в пункті 2 методики (таблиці 3.9.-3.10.) [18]. Для зручності відображення стовпчики та рядки таблиць змінені місцями.

Перша таблиця для оптимізації моделей

№ функції	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$X_1 \vee X_2 \vee X_3$																	
$X_1 X_2 \vee X_3 \vee X_4 X_5 X_6 X_7$																	
$X_1 \vee X_2 \vee X_4 X_5 X_6 X_7$																	
$X_1 X_2 X_3 \vee X_4 \vee X_5$																	
$X_1 X_2 X_3 \vee X_6 \vee X_7$																	
$X_1 X_2 X_3 \vee X_4 X_5 \vee X_6 X_7$																	
$X_4 \vee X_5 \vee X_6 X_7$																	
$X_4 X_5 \vee X_6 \vee X_7$																	
$X_3 \vee (X_1 \vee X_2) (X_1 X_2 \vee X_4 X_5 X_6 X_7)$																	
$X_1 \vee X_2 \vee X_3 X_4 X_5 X_6 X_7$																	
$X_4 X_5 X_6 X_7 \vee (X_1 \vee X_2) (X_1 X_2 \vee X_3)$																	
$x_1 x_2 x_3 \vee (x_4 \vee x_5) (x_6 \vee x_7) (x_4 x_5 \vee x_6 x_7)$																	
$X_4 \vee X_5 \vee X_1 X_2 X_3 X_6 X_7$																	
$X_6 \vee X_7 \vee X_1 X_2 X_3 X_4 X_5$																	
$X_6 X_7 \vee (X_1 X_2 X_3 \vee X_4 X_5) (X_4 \vee X_5)$																	
$X_4 X_5 \vee (X_1 X_2 X_3 \vee X_6 X_7) (X_6 \vee X_7)$																	
$X_1 X_2 X_3 \vee (X_4 \vee X_5) (X_6 \vee X_7)$																	
$X_1 X_2 X_3 X_4 X_5 X_6 X_7$																	
0 0 0 1 1 1 1	*								*	*							
0 0 1 0 1 1 1			*						*	*	*						
0 1 0 0 1 1 1		*							*		*						
1 0 0 0 1 1 1		*							*		*						
0 0 1 1 0 1 1			*							*	*						
0 1 0 1 0 1 1		*							*		*						
1 0 0 1 0 1 1		*							*		*						
0 1 1 0 0 1 1				*								*	*			*	
1 0 1 0 0 1 1				*								*	*			*	
1 1 0 0 0 1 1				*								*	*			*	
0 0 1 1 1 0 1			*							*	*						
0 1 0 1 1 0 1		*							*		*						
1 0 0 1 1 0 1		*							*		*						
0 1 1 0 1 0 1						*						*			*	*	
1 0 1 0 1 0 1						*						*			*	*	
1 1 0 0 1 0 1						*						*			*	*	
0 1 1 1 0 0 1						*						*			*	*	
1 0 1 1 0 0 1						*						*			*	*	
1 1 0 1 0 0 1						*						*			*	*	
1 1 1 0 0 0 1							*						*		*		
0 0 1 1 1 1 0			*							*	*						
0 1 0 1 1 1 0		*							*		*						
1 0 0 1 1 1 0		*							*		*						
0 1 1 0 1 1 0						*						*			*	*	
1 0 1 0 1 1 0						*						*			*	*	
1 1 0 0 1 1 0						*						*			*	*	
0 1 1 1 0 1 0						*						*			*	*	
1 0 1 1 0 1 0						*						*			*	*	
1 1 0 1 0 1 0						*						*			*	*	
1 1 1 0 0 1 0							*						*		*		
0 1 1 1 1 0 0					*							*		*			*
1 0 1 1 1 0 0					*							*		*			*
1 1 0 1 1 0 0					*							*		*			*
1 1 1 0 1 0 0								*						*		*	
1 1 1 1 0 0 0								*						*		*	

Друга таблиця для оптимізації моделей

№ ф у н к ц і ї	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$x_1 x_2 x_3 x_4 x_5 x_6 x_7$	$x_1 \vee x_2 \vee x_3$	$x_1 x_2 \vee x_3 \vee x_4 x_5 x_6 x_7$	$x_1 \vee x_2 \vee x_4 x_5 x_6 x_7$	$x_1 x_2 x_3 \vee x_4 \vee x_5$	$x_1 x_2 x_3 \vee x_6 \vee x_7$	$x_1 x_2 x_3 \vee x_4 x_5 \vee x_6 x_7$	$x_4 \vee x_5 \vee x_6 x_7$	$x_4 x_5 \vee x_6 \vee x_7$	$x_3 \vee (x_1 \vee x_2) (x_1 x_2 \vee x_4 x_5 x_6 x_7)$	$x_1 \vee x_2 \vee x_3 x_4 x_5 x_6 x_7$	$x_4 x_5 x_6 x_7 \vee (x_1 \vee x_2) (x_1 x_2 \vee x_3)$	$x_1 x_2 x_3 \vee (x_4 \vee x_5) (x_6 \vee x_7)$	$x_4 \vee x_5 \vee x_1 x_2 x_3 x_6 x_7$	$x_6 \vee x_7 \vee x_1 x_2 x_3 x_4 x_5$	$x_6 x_7 \vee (x_1 x_2 x_3 \vee x_4 x_5) (x_4 \vee x_5)$	$x_4 x_5 \vee (x_1 x_2 x_3 \vee x_6 x_7) (x_6 \vee x_7)$	$x_1 x_2 x_3 \vee (x_4 \vee x_5) (x_6 \vee x_7)$
0 0 0 1 1 1 1	*								*	*							
0 0 1 0 1 1 1			*							*	*						
0 1 0 0 1 1 1		*							*		*						
1 0 0 0 1 1 1		*							*		*						
0 0 1 1 0 1 1			*							*	*						
0 1 0 1 0 1 1		*							*		*						
1 0 0 1 0 1 1		*							*		*						
0 1 1 0 0 1 1				*								*	*				*
1 0 1 0 0 1 1				*								*	*				*
1 1 0 0 0 1 1				*								*	*				*
0 0 1 1 1 0 1			*							*	*						
0 1 0 1 1 0 1		*							*		*						
1 0 0 1 1 0 1		*							*		*						
0 1 1 0 1 0 1						*						*			*	*	
1 0 1 0 1 0 1						*						*			*	*	
1 1 0 0 1 0 1						*						*			*	*	
0 1 1 1 0 0 1						*						*			*	*	
1 0 1 1 0 0 1						*						*			*	*	
1 1 0 1 0 0 1						*						*			*	*	
1 1 1 0 0 0 1							*						*		*		
0 0 1 1 1 1 0			*							*	*						
0 1 0 1 1 1 0		*							*		*						
1 0 0 1 1 1 0		*							*		*						
0 1 1 0 1 1 0						*						*			*	*	
1 0 1 0 1 1 0						*						*			*	*	
1 1 0 0 1 1 0						*						*			*	*	
0 1 1 1 0 1 0						*						*			*	*	
1 0 1 1 0 1 0						*						*			*	*	
1 1 0 1 0 1 0						*						*			*	*	
1 1 1 0 0 1 0							*						*		*		
0 1 1 1 1 0 0					*							*		*			*
1 0 1 1 1 0 0					*							*		*			*
1 1 0 1 1 0 0					*							*		*			*
1 1 1 0 1 0 0								*						*		*	
1 1 1 1 0 0 0								*						*		*	

Виходячи зі вказаних вище критеріїв, необхідно обрати функції 3.35.

$$\begin{aligned}
 & x_1 \vee x_2 \vee x_3 \\
 & x_4 \vee x_5 \vee x_6 x_7 \\
 & x_6 \vee x_7 \vee x_4 x_5 \\
 & x_4 x_5 x_6 x_7 \vee (x_1 \vee x_2)(x_1 x_2 \vee x_3) \\
 & x_1 x_2 x_3 \vee (x_4 \vee x_5)(x_6 \vee x_7)(x_4 x_5 \vee x_6 x_7)
 \end{aligned} \tag{3.35}$$

Недоліком такої методики є велика обчислювальна складність. Це є наслідком того, що для отримання функцій оптимізованої за обраним критерієм моделі необхідно виконати перебір усіх можливих комбінацій функцій та усіх можливих комбінацій їх склеювання. Таким чином великий вплив на час виконання такої методики мають кількість модулів системи n та відмовостійкість системи m .

Дана методика таким чином надає можливість виконувати оптимізацію *GL*-моделей за декількома критеріями.

3.4. Критерії оптимальності *GL*-моделей

Як вже зазначалося раніше, *GL*-модель може мати значно простішу структуру, ніж структура реальної ВБС. Це є наслідком перенесення складності структури системи на реберні функції.

Одним з зазначених вище критеріїв якості моделі є складність її комп'ютерного представлення. Для приблизної оцінки зробимо такі припущення:

- складність представлення ребра графу дорівнює складності представлення двомісної булевої функції між термами;
- складність операції заперечення дорівнює нулю.

Тоді критерій складності комп'ютерного представлення моделі математично можна описати за допомогою виразу 3.36.

$$S = R + \sum_{i=1}^R S_i \tag{3.36}$$

В формулі 3.36 R – кількість ребер графу моделі системи, а S_i – складність представлення булевої функції i -ого ребра відповідно до кількості двомісних операцій.

За критерієм 3.36 можна порівняти різноманітні GL-моделі. Для GL-моделей циклічного типу їх складність комп'ютерного представлення надана в таблиці 3.11.

Таблиця 3.11

Складність комп'ютерного представлення GL-моделей циклічного типу з різними значенням n та m

$\begin{matrix} n \\ m \end{matrix}$	4	5	6	7	8	9	10	11	12	13	14	15	16
2	8	12	16	20	24	29	34	39	44	49	54	59	64
3	*	17	28	41	56	76	98	122	148	176	206	238	272
4	*	*	32	56	88	137	198	269	352	443	546	659	784
5	*	*	*	50	96	179	300	454	652	883	1164	1490	1872
6	*	*	*	*	72	169	340	591	952	1403	1994	2725	3632
7	*	*	*	*	*	108	278	586	1092	1779	2760	4077	5824
8	*	*	*	*	*	*	158	430	964	1788	3088	5010	7764
9	*	*	*	*	*	*	*	225	648	1414	2780	5038	8320
10	*	*	*	*	*	*	*	*	312	852	1982	4119	7814
11	*	*	*	*	*	*	*	*	*	368	1084	2680	5862
12	*	*	*	*	*	*	*	*	*	*	428	1344	3520
13	*	*	*	*	*	*	*	*	*	*	*	492	1632
14	*	*	*	*	*	*	*	*	*	*	*	*	560

Зокрема складність 2-відмовостійких моделей систем можна обчислити за формулою 3.37

$$S = n\left(\left\lfloor \frac{n-1}{2} \right\rfloor + 1\right) \quad (3.37)$$

Критерій, що використовується для оптимізації моделей в даній роботі, є критерієм мінімуму складності реберних функцій. Він дуже схожий на критерій складності комп'ютерного представлення моделі, проте відображає лише загальну складність реберних функцій моделі та може бути обчисленим за допомогою виразу 3.38.

$$S = \sum_{i=1}^R S_i \quad (3.38)$$

За критерієм оптимальності 3.38 можна порівняти різноманітні GL-моделі. Для GL-моделей циклічного типу складність їх реберних функцій надана в таблиці 3.12.

Таблиця 3.12

Складність реберних функцій GL-моделей циклічного типу з різними значеннями n та m

$n \backslash m$	2	3	4	5	6
4	5	*	*	*	*
5	8	13	*	*	*
6	11	22	22	*	*
7	14	32	46	42	*
8	17	46	73	82	66

Причиною вибору такого критерію може бути використання паралельних обчислень для одночасної оцінки значень усіх реберних функцій моделі. У

зв'язку з тим, що реальні системи здебільшого мають дуже велику кількість модулів, а моделі таких систем мають відповідно велику кількість реберних функцій, така задача є дуже актуальною в наш час.

Збільшується не тільки кількість реберних функцій, але й кількість можливих векторів стану системи, на яких необхідно тестувати модель. Таким чином збільшення кількості модулів впливає на збільшення кількості обчислень майже на порядок.

В таких умовах, коли значення реберних функцій, які вони набувають на векторах стану, є незалежними одне від одного, використання паралельних обчислень часто є необхідним для зменшення часу виконання експериментів над моделями.

3.5. Спосіб початкового розбиття елементів системи групи для алгоритму послідовного поділу на групи для спрощення перетворення базових моделей в небазові

Алгоритм послідовного поділу на групи є універсальним, легким для реалізації алгоритмом, проте питанню визначення ефективного поділу елементів системи на групи не було приділено достатньо уваги.

Але зрозуміло, що саме спосіб поділу елементів впливає на остаточний вигляд реберних функцій моделі. Існує декілька варіантів розбиття елементів системи на групи. Визначено, що оптимальним варіантом розбиття елементів на рівні або майже рівні за потужністю підмножини. Проте не є очевидним, як саме потрібно групувати елементи системи за цими групами. Тобто тільки для 3 відмовостійкої системи, що складається з 7 модулів, можливі такі варіанти початкового поділу елементів системи на групи:

$$\begin{aligned} &\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7\} \\ &\{x_1, x_2, x_3, x_5\}, \{x_4, x_6, x_7\} \\ &\{x_1, x_2, x_3, x_6\}, \{x_4, x_5, x_7\} \\ &\{x_1, x_2, x_3, x_7\}, \{x_4, x_5, x_6\} \\ &\{x_1, x_2, x_5, x_4\}, \{x_3, x_6, x_7\} \\ &\{x_1, x_2, x_6, x_4\}, \{x_3, x_5, x_7\} \end{aligned} \tag{3.39}$$

$$\{x_1, x_2, x_7, x_4\}, \{x_3, x_5, x_6\}$$

$$\{x_1, x_5, x_3, x_4\}, \{x_7, x_2, x_6\}$$

...

$$\{x_4, x_5, x_6, x_1\}, \{x_3, x_2, x_7\}$$

$$\{x_4, x_5, x_6, x_7\}, \{x_1, x_2, x_3\}$$

В виразі 3.39 наведено лише декілька прикладів. Загальна кількість таких можливих варіантів дорівнює 35. І це число збільшується не тільки зі збільшенням загальної кількості модулів в системі, але й за можливим нехтуванням наступних умов:

- розбиття елементів системи на рівні або майже рівні за потужностями підмножини;
- розгляд тільки початкової ітерації розбиття елементів;
- розбиття елементів системи на 2 групи в кожній ітерації.

Таким чином необхідною є розробка методики визначення елементів системи, які повинні знаходитися в одній підмножині для подальшого розподілу відмов.

В даній магістерській дисертації пропонується методика початкового поділу елементів системи на підмножини в залежності від частоти набування ними нульового значення в векторах стану, які необхідно блокувати [19-20].

Розглянемо реберні функції моделі $K(3, 7)$ системи, що були отримані за допомогою алгоритму послідовного поділу за початкового розбиття елементів системи на підмножини $\{x_1, x_2, x_3\}, \{x_4, x_5, x_6, x_7\}$ (3.40) та на підмножини $\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7\}$ (3.41).

$$f_1 = x_1 \vee x_2 \vee x_3$$

$$f_2 = x_1 \vee x_2 \vee x_4 x_5 x_6 x_7$$

$$f_3 = x_1 x_2 \vee x_3 \vee x_4 x_5 x_6 x_7$$

$$f_4 = x_4 \vee x_5 \vee x_1 x_2 x_3 \tag{3.40}$$

$$f_5 = x_6 \vee x_7 \vee x_1 x_2 x_3$$

$$f_6 = x_4 x_5 \vee x_6 x_7 \vee x_1 x_2 x_3$$

$$f_7 = x_4 \vee x_5 \vee x_6 x_7$$

$$\begin{aligned}
f_8 &= x_6 \vee x_7 \vee x_4 x_5 \\
f_1 &= x_5 \vee x_6 \vee x_7 \\
f_2 &= x_5 \vee x_6 \vee x_1 x_2 x_3 x_4 \\
f_3 &= x_5 x_6 \vee x_7 \vee x_1 x_2 x_3 x_4 \\
f_4 &= x_1 \vee x_2 \vee x_5 x_6 x_7 \\
f_5 &= x_3 \vee x_4 \vee x_5 x_6 x_7 \\
f_6 &= x_1 x_2 \vee x_3 x_4 \vee x_5 x_6 x_7 \\
f_7 &= x_1 \vee x_2 \vee x_3 x_4 \\
f_8 &= x_3 \vee x_4 \vee x_1 x_2
\end{aligned} \tag{3.41}$$

Модель, реберні функції якої наведені в 3.40 відповідає початковому поділу елементів системи за способом, запропонованим вище, у випадку, коли необхідно відобразити на моделі підвищення відмовостійкості системи за рахунок блокування вектора 3.42.

$$\begin{array}{ccccccc}
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\
1 & 1 & 1 & 0 & 0 & 0 & 0
\end{array} \tag{3.42}$$

Для моделі з функціями 3.41 даний вектор повинен мати вигляд 3.43.

$$\begin{array}{ccccccc}
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\
0 & 0 & 0 & 0 & 1 & 1 & 1
\end{array} \tag{3.43}$$

Якщо уважно подивитись на наведені вище реберні функції, то можна помітити, що у випадку винесення нульових елементів вектора, що необхідно блокувати, в окрему групу, будуть існувати реберні функції, що складаються лише з цих елементів. Тобто у випадку блокування вектора 3.42. в моделі 3.40 такими є реберні функції, наведені в 3.44., а у випадку блокування 3.43. – функції 3.45.

$$\begin{aligned}
&x_4 \vee x_5 \vee x_6 x_7 \\
&x_6 \vee x_7 \vee x_4 x_5
\end{aligned} \tag{3.44}$$

$$\begin{aligned}
&x_1 \vee x_2 \vee x_3 x_4 \\
&x_3 \vee x_4 \vee x_1 x_2
\end{aligned} \tag{3.45}$$

Ця властивість є дуже важливою для запропонованого способу модифікації базових моделей в небазові, тому на даному етапі необхідно довести справедливості даного твердження.

Розглянемо деяку базову модель $K(n, m)$ системи, яку необхідно перетворити в небазову за допомогою відображення підвищення ступеня відмовостійкості після блокування деякого вектора з $m + k$ нульовими компонентами. Зазначений вектор буде мати вигляд 3.46, де b_i – це компонента вектора, що набуває нульового значення, а компонента a_i набуває відповідно значення 1.

$$b_2, b_2, \dots b_{m+k}, a_{m+k+1}, \dots a_n \quad (3.46)$$

Тоді за умови винесення елементів системи b_i в окрему групу будуть існувати такі комбінації розподілення відмов $K_{|n_j|}^{m_{ij}}$, де всі відмови будуть відповідати елементам тільки цієї групи тому, що кількість елементів в даній підмножині n_j буде дорівнювати $m + k$, що більше загальної кількості відмов m , які необхідно розподілити.

Візуально цю закономірність відображено на рис. 3.6. Тут відмови позначені червоними квадратами, а елементи системи – білими.

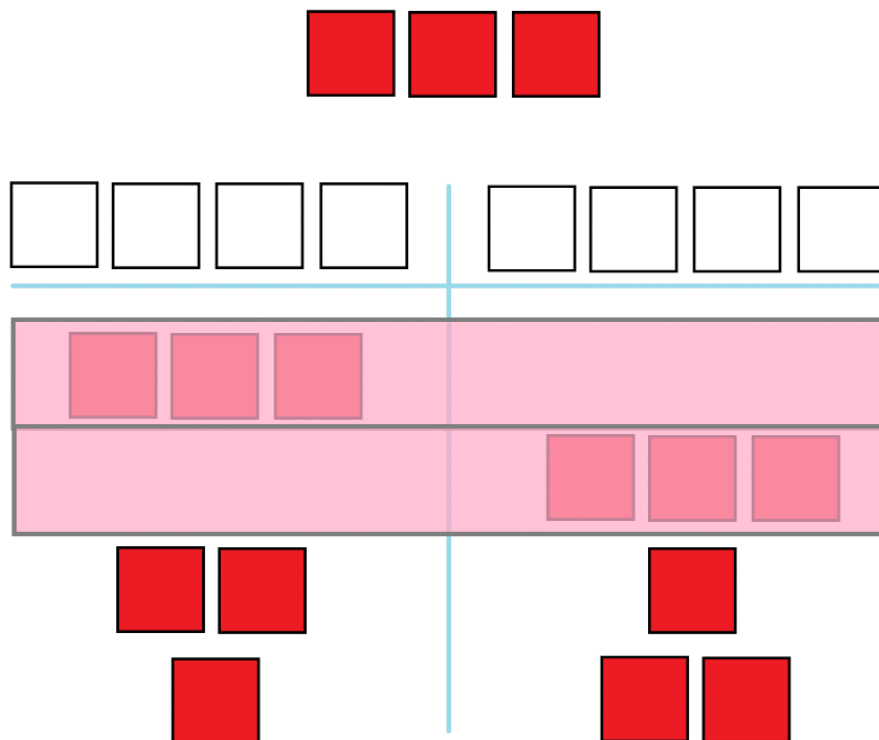


Рис. 3.6. Розподіл відмов за підсистемами у випадку винесення елементів системи b_i в окрему групу

Дана властивість цих реберних функцій дозволяє спростити процес відображення підвищення ступеня відмовостійкості на базових моделях, який більш детально буде розглянуто в наступному розділі.

3.6. Запропонований спосіб відображення підвищення ступеня відмовостійкості

Дослідження показали, що у випадку поділу елементів системи на групи за способом, запропонованим в пункті 3.5., достатньою умовою коректного відображення підвищення ступеня відмовостійкості системи на GL-моделях є додання до реберних функцій, що набувають значення 0 кон'юнкції елементів, що набувають значення 1 на векторі стану, який необхідно блокувати. Доведемо це положення.

Як було показано в пункті 3.5., за умови запропонованого поділу елементів системи на групи будуть існувати реберні функції, що залежать тільки від стану модулів системи, які відмовляють на векторі 3.46. Таким чином, за появи цього вектора стану вказані реберні функції відмовляють. Це в свою чергу позначає, що саме ці функції необхідно модифікувати (як і у випадку стандартного алгоритму додавання конститuentи).

Модифікована реберна функція повинна набувати значення 1 на заданому векторі стану, проте не змінювати свою поведінку за появи інших векторів. Реберні функції, отримані за алгоритмом послідовного поділу на групи, знаходяться в ДНФ, та можуть містити від одного до $m + k$ нульових компонентів вектора.

В загальному вигляді модифіковані за вказаним правилом функції будуть забезпечувати роботоздатність системи за умови появи наступного вектора стану:

$$\begin{array}{ccccccc} b_2 & b_2 & \dots & b_{m+k} & a_{m+k+1} & \dots & a_n \\ x & x & x & x & 1 & 1 & 1 \end{array} \quad (3.47)$$

У векторі стану 3.47 поміткою x позначений невизначений стан компоненти, що залежить від вигляду функції, що було модифіковано. Тоді зрозуміло, що запропонований спосіб є справедливим за модифікації функції з 3.48, яка після перетворення буде мати вигляд 3.49

$$b_2 \vee b_2 \vee \dots \vee b_{m+k} \quad (3.48)$$

$$a_{m+k+1}, \dots, a_n \vee b_2 \vee b_2 \vee \dots \vee b_{m+k} \quad (3.49)$$

Така модифікація функції призводить до того, що за появи вектора стану 3.50 вона не буде видалена із графу та граф залишиться зв'язним, тобто для такого випадку даний спосіб є справедливим.

$$\begin{array}{ccccccc} b_2 & b_2 & \dots & b_{m+k} & a_{m+k+1} & \dots & a_n \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array} \quad (3.50)$$

Проте зрозуміло, що не обов'язково функції, які необхідно модифікувати, будуть мати такий вигляд. Розглянемо два інших випадки:

1. Відображення підвищення ступеня відмовостійкості на моделях за рахунок блокування вектора, що містить $m + 1$ нульових компонент (3.51).

$$\begin{array}{ccccccc} b_1 & b_2 & \dots & b_{m+1} & a_{m+2} & \dots & a_n \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array} \quad (3.51)$$

В такому випадку в залежності від функції для модифікації будуть заблоковані комбінації відмов, що містять від 0 до $m + 1$ нульових компонент з підмножини b . Тобто буде заблоковано або вектор 3.51. (що і є поставленим завданням), або $< m$ відмов компонентів, що не впливає на функціонування m -відмовостійкої системи.

2. Відображення підвищення ступеня відмовостійкості за рахунок блокування вектора, що містить більше, ніж $m + 1$ нульових компонент. Тоді в найгіршому випадку будуть заблоковані усі вектори стану форми 3.47., що є абсолютно логічним з точки зору проектувальника даної системи.

Наприклад, якщо 2-відмовостійка системи залишається роботоздатною за відмови компонентів $\{x_4, x_5, x_6, x_7\}$, то відмова компонентів $\{x_5, x_6, x_7\}$ також відповідно не повинна спричиняти відмову системи.

Наведемо приклад використання даного способу для блокування вектора стану з $m + 1$ нулем.

Нехай необхідно створити модель 2-відмовостійкої системи, що містить 8 модулів та залишається роботоздатною за появи вектора стану 3.52.

$$\begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{array} \quad (3.52)$$

В такому випадку, як вказано в розділі 3.5. даної магістерської дисертації, елементи системи необхідно розбити на підмножини $\{x_1, x_2, x_3, x_4, x_5\}, \{x_6, x_7, x_8\}$ в першій ітерації. Отримані для даної моделі реберні функції наведені в 3.53.

$$\begin{aligned} f_1 &= x_1 x_2 x_3 x_4 x_5 \vee x_6 x_7 x_8 \\ f_2 &= x_1 \vee x_2 \\ f_3 &= x_1 x_2 \vee x_3 \\ f_4 &= x_1 x_2 x_3 \vee x_4 x_5 \\ f_5 &= x_4 \vee x_5 \\ f_6 &= x_6 \vee x_7 \\ f_7 &= x_6 x_7 \vee x_8 \end{aligned} \quad (3.53)$$

В даній моделі за появи вектора 3.52 функції f_6 та f_7 набувають нульового значення. Для прикладу модифікуємо функцію f_7 за запропонованим способом. Результат модифікації наведено в 3.54.

$$f_7 = x_6 x_7 \vee x_8 \vee x_1 x_2 x_3 x_4 x_5 \quad (3.54)$$

Порівняємо результат відображення підвищення ступеня відмовостійкості за допомогою даного способу та результат, отриманий за допомогою оригінального алгоритму модифікації конститuentoю. В останньому випадку нам було б необхідно перетворити або функцію f_6 до вигляду 3.55, або функцію f_7 до вигляду 3.56.

$$f_7 = x_6 \vee x_7 \vee \overline{x_6} \overline{x_7} \overline{x_8} x_1 x_2 x_3 x_4 x_5 \quad (3.55)$$

$$f_7 = x_6 x_7 \vee x_8 \vee \overline{x_6} \overline{x_7} \overline{x_8} x_1 x_2 x_3 x_4 x_5 \quad (3.56)$$

Розглянемо поведінку функції f_7 після перетворення за запропонованим способом. Модифікація впливає на функціонування системи за появи векторів стану вигляду 3.57.

$$\begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6/x_7 & x_8 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{array} \quad (3.57)$$

Таким чином функція буде набувати значення 1 не тільки за появи вектора стану 3.52., але й за появи векторів стану 3.58. та 3.59.

$$\begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{array} \quad (3.58)$$

$$\begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \end{array} \quad (3.59)$$

Але у зв'язку з тим, що дана системи є 2-відмовостійкою, така зміна не є важливою, бо граф у будь-якому випадку повинен зберігати зв'язність, і кількість ребер, що він втрачає після модифікації, не впливає на відображення стану системи.

Також необхідно зазначити, що даний спосіб може бути використаним для блокування декількох векторів за умови відповідного поділу елементів системи на групи в кожній ітерації, проте необхідно визначити умови ефективного його використання в такому випадку.

3.7. Аналіз способу перетворення GL-моделей в небазові шляхом відображення підвищення ступеня відмовостійкості.

Зрозуміло, що запропонований спосіб відображення підвищення ступеня відмовостійкості дозволяє мінімізувати модель за критерієм складності реберних функцій, бо в такому випадку не потрібно додавати цілу конституюнту, як в оригінальному способі, а тільки її частину, що складається з компонентів вектора, що набувають значення 1 на векторі, що необхідно блокувати.

Проте слід розуміти, що у даного способу існують і свої обмеження. Перший з них є наслідком обмеження алгоритму послідовного поділу на групи, яке вказує на те, що для отримання оптимальних моделей необхідно ділити елементи системи на кожній ітерації на рівні або майже рівні за потужністю підмножини.

Згадаємо, що за запропонованим методом відображення підвищення ступеня відмовостійкості необхідно ділити модулі системи в залежності від частоти набування ними 0 в векторах стану системи, які необхідно блокувати. Тоді зрозуміло, що умовою ефективного використання способу є блокування векторів стану, де кількість нульових компонент приблизно дорівнює половині кількості модулів системи (3.60).

$$m + k \approx \frac{n}{2} \quad (3.60)$$

В даній магістерській дисертації було досліджено використання даного способу лише для відображення підвищення ступеня відмовостійкості систем, тому темою наступних наукових досліджень може слугувати адаптація запропонованого способу для відображення зниження ступеня відмовостійкості.

Висновки розділу 3

В даному розділі розроблено спосіб перетворення GL-моделей в небазові за рахунок відображення підвищення ступеня відмовостійкості системи на основі алгоритму послідовного поділу на групи, що є універсальним для будь-яких значень m та n . Запропонований спосіб дозволяє отримати коротші реберні функції для моделі, ніж за умови використання способу з додавання цілої конститuentи до ребер, що набувають значення 0. Спосіб передбачає поділ елементів системи в першій ітерації за частотою набування відповідними компонентами значення 0 в векторах стану, які необхідно блокувати.

Запропонований спосіб має ті ж обмеження ефективного використання, що має і алгоритм послідовного поділу елементів системи на групи: поділ елементів системи повинен виконуватися на рівні або майже рівні за потужністю підмножини. Це в свою чергу означає, що мінімальні за довжиною реберні функції за допомогою запропонованого способу можна отримати, якщо кількість нульових компонент в векторах стану, що необхідно блокувати, приблизно дорівнює половині загальної кількості модулів в системі.

ВИСНОВКИ

В дисертації розглянуто проблему обчислення ймовірності безвідмовної роботи ВБС статистичним методом за допомогою використання спеціальних моделей для відображення поведінки системи в потоці відмов. Встановлено, що GL-моделі за рахунок простоти своєї структури дозволяють провести більшу кількість експериментів над ним за фіксований час, ніж інші розглянуті моделі, та у зв'язку з цим точніше розрахувати ймовірність безвідмовної роботи системи.

В роботі проаналізовано різні алгоритми формування GL-моделей. Встановлено, що порівняно з іншими підходами алгоритм послідовного поділу елементів системи на групи є універсальним та більш гнучким для модифікації. Тому саме на цьому алгоритмі було засновано запропонований спосіб перетворення GL-моделей в небазові за рахунок відображення підвищення на моделях ступеня відмовостійкості реальної ВБС. Інші існуючі способи відображення зміни ступеня відмовостійкості за допомогою модифікації графу та модифікації реберних функцій було також наведено.

Були розглянуті та проаналізовані різні критерії оптимальності GL-моделей. На основі аналізу було обрано критерій мінімізації реберних функцій для оцінки використання запропонованого способу у порівнянні з оригінальним способом перетворення GL-моделей в небазові, що заснований на модифікації реберних функцій за допомогою додавання конститuentи.

Було проаналізовано та доведено справедливність використання способу для відображення на моделях підвищення ступеня відмовостійкості за рахунок блокування додаткових векторів стану, що містять $m + 1$ та більше нульових компонент.

Аналіз запропонованого способу показав, що:

- для використання способу необхідною умовою є розбиття елементів системи на підмножини в першій ітерації алгоритму послідовного поділу на групи в залежності від частоти набування ними нульового значення в векторах стану, що необхідно блокувати;
- обмеження ефективного використання алгоритму послідовного поділу на групи (поділ елементів системи на приблизно рівні за потужністю

підмножини) також є актуальним для запропонованого способу, що у випадку з багатьма векторами стану, які необхідно блокувати, є складним до виконання.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Романкевич А. М. Графо-логические модели для анализа сложных отказоустойчивых вычислительных систем/ Романкевич А. М., Карачун Л. Ф., Романкевич В. А. // Электронное моделирование.– 2001.–т.23, № 1.– С. 102-111.
2. Шкляр В. Н. Надежность систем управления: учебное пособие / В. Н. Шкляр; Томский политехнический университет. – Томск: Издательство Томского политехнического университета, 2009. – 126 стр.
3. Павловская, О. О. Основы теории надежности: учебное пособие / О.О. Павловская. – Челябинск: Издательский центр ЮУрГУ, 2013. – 94 с.
4. Федотов, А. В. Основы теории надежности и технической диагностики: конспект лекций / А. В. Федотов, Н. Г. Скабкин. – Омск : Изд-во ОмГТУ, 2010. – 64 с.
5. Коваленко А.Г. Отказоустойчивые микропроцессорные системы / В.В. Гула – К.: Техніка, 1986 – 148с.
6. Романкевич О. М. Аналіз відмовостійких багатопроцесорних систем на основі графо-логічних моделей нециклічного типу/ О. М. Романкевич, В. О. Романкевич, О. В. Богуславський, Ал Шбул Рабах. // Вісник Технологічного університету Поділля. – 2002. – т.2, №3. – с. 30-33.
7. Райншке К. Оценка надежности систем с использованием графов / К. Райншке, И. А. Ушаков/Под ред. И. А. Ушакова.– М.: Радио и связь, 1988.–208с.
8. Kuo W., Zuo Ming J. The k-out-of-n system model [Електронний ресурс] / Kuo W., Zuo Ming J. – [50 p.]. – Режим доступу: <http://www.ewp.rpi.edu/hartford/~ernesto/S2008/SMRE/Papers/Kuo-Zuo-koon.pdf>. – Назва з екрану.
9. Barlow R.E. Computing k-out-of-n system reliability/ R. E. Barlow, K. D. Heidtmann // IEEE Trans. Reliab. – 1984. – Vol. 33. – p.322-323.

10. Endharta A. J. Preventive maintenance policy for linear consecutive k-out-of-n: F system [Електронний ресурс] / A. J. Endharta, W. Y. Yun, Yamamoto H. – [13 p]. – Режим доступу: http://www.orsj.or.jp/~archive/pdf/e_mag/Vol.59_04_334.pdf. – Назва з екрану.
11. Романкевич В. А. Об одной модели поведения отказоустойчивой многопроцессорной системы / В. А. Романкевич // Радиоэлектроника и информатика. – 1999. – №1. – С.75-76.
12. Романкевич О. М. До питання побудови моделі поведінки багатомодульних систем / О. М. Романкевич, Л. Ф. Карачун, В. О. Романкевич // Наукові вісті НТУУ “КПІ”. – 1998. – №1. – с. 38-40.
13. Романкевич В. А. Об одном способе построения GL-моделей отказоустойчивых многопроцессорных систем / В. А. Романкевич // Радіоелектронні і комп’ютерні системи.–2006.– № 7.– С. 47-51.
14. Самофалов К. Г. Прикладная теория цифровых автоматов / К. Г. Самофалов, А. М. Романкевич, В. Н. Валуйский, Ю. С. Каневский, М. М. Пиневиц – К. :Вища школа, 1987. – 375с.
15. Wegener I. The complexity of Boolean functions / Wegener I. – Stuttgart: Teubner, 1987. – 458 p.
16. Винокуров С. Ф. Избранные вопросы теории булевых функций / С. Ф. Винокуров, Н. А. Перязев, ред. – М.: Физматлит, 2001.- 192с.
17. Романкевич В. А. GL-модель поведения отказоустойчивых многопроцессорных систем с минимальным числом теряемых рёбер/ В. А. Романкевич, Е. Р. Потапова, Бахтари Хедаятоллах, В. В. Назаренко // Вісник НТУУ “КПІ”.– Інформатика, управління та ОТ. – 2006. – №45. – С.93-100.
18. Романкевич А. М. Об одном способе оптимизации моделей поведения отказоустойчивости многопроцессорных систем/ А. М. Романкевич, В.

- А. Романкевич, Бахтари Хедаятоллах // Вісник НТУУ “КПІ”. – Інформатика, управління та ОТ. – 2008. – №45. – 5 с.
19. Романкевич В. А. Об одном способе формирования реберных функций GL-модели/ В. А. Романкевич, М. О. Малышева, И. К. Примаков // Системный анализ и информационные технологии: материалы 18-й Международной научно-технической конференции SAIT 2016, Киев, 30 мая – 2 июня 2016 г. / УНК “ИПСА” НТУУ “КПИ”. – К.: УНК “ИПСА” НТУУ “КПИ”, 2016. – С. 406–407.
20. Романкевич В. О. Спосіб перетворення GL-моделей за критерієм мінімуму складності реберних функцій/ В. О. Романкевич, М. О. Малишева // Прикладна математика та комп’ютинг. ПМК, 2018 : десята наук. конф. магістрантів та аспірантів, Київ, 21 – 23 березня 2018 р. : зб. тез. доп. – К.: Просвіта, 2018. – С. 132–136.

Додаток 1. Копії графічних матеріалів

Алгоритм послідовного поділу елементів системи на групи для формування GL-моделей

$$\left\{ \begin{array}{cccccc} K_{|n_1|}^{m_{11}} & \vee & K_{|n_2|}^{m_{12}} & \vee & \dots & \vee & K_{|n_t|}^{m_{1t}} \\ K_{|n_1|}^{m_{21}} & \vee & K_{|n_2|}^{m_{22}} & \vee & \dots & \vee & K_{|n_t|}^{m_{2t}} \\ \dots & & \dots & & \dots & & \dots \\ K_{|n_1|}^{m_{m1}} & \vee & K_{|n_2|}^{m_{m2}} & \vee & \dots & \vee & K_{|n_t|}^{m_{mt}} \end{array} \right.$$

$$K_{|n_j|}^0 = 0$$

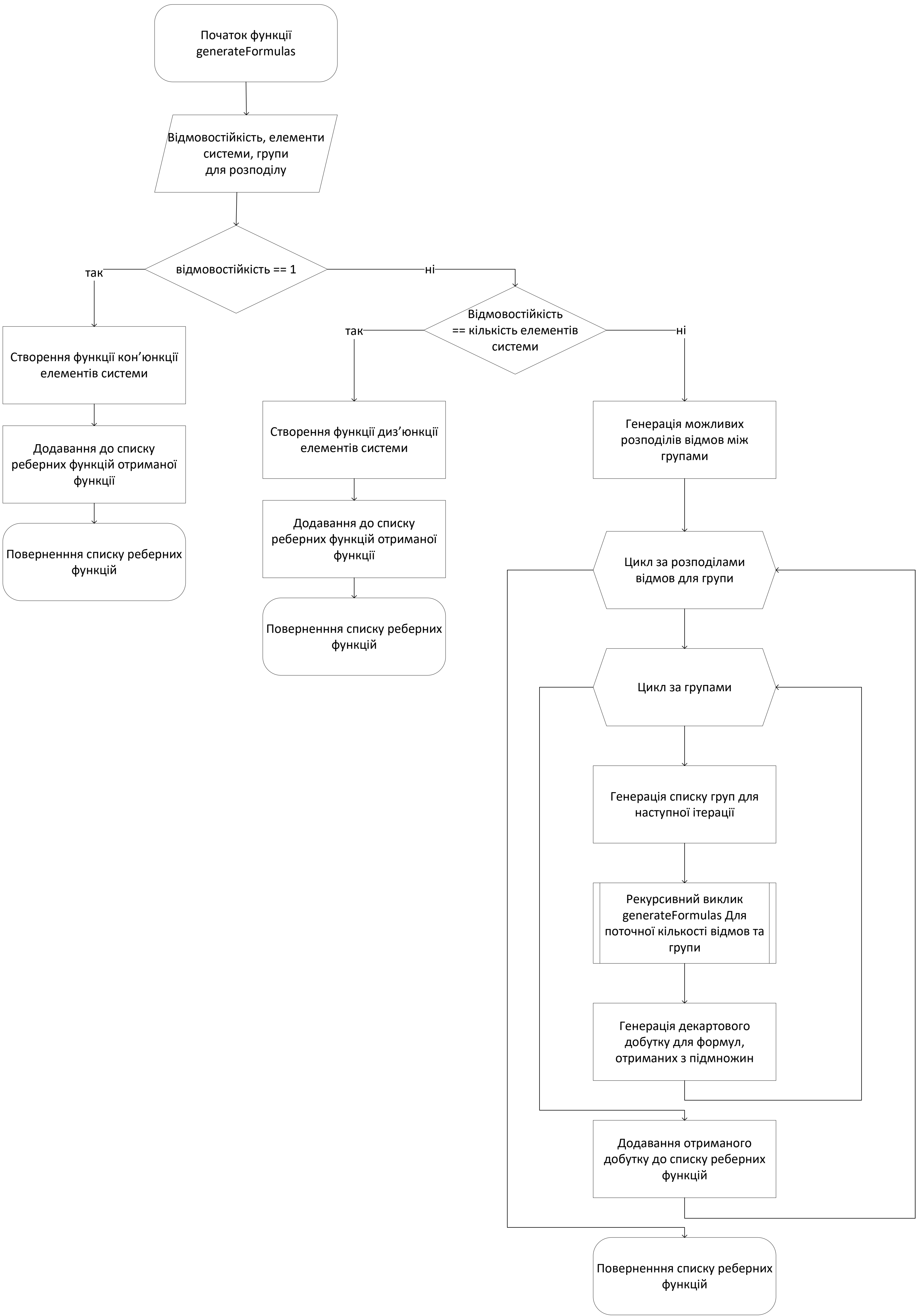
$$K_{|n_j|}^1 = \bigwedge_{k \in n_j} x_k$$

$$K_{|n_j|}^{m_j} = \bigvee_{k \in n_j} x_k$$

Спосіб перетворення GL-моделей за
критерієм мінімуму складності реберних
функцій

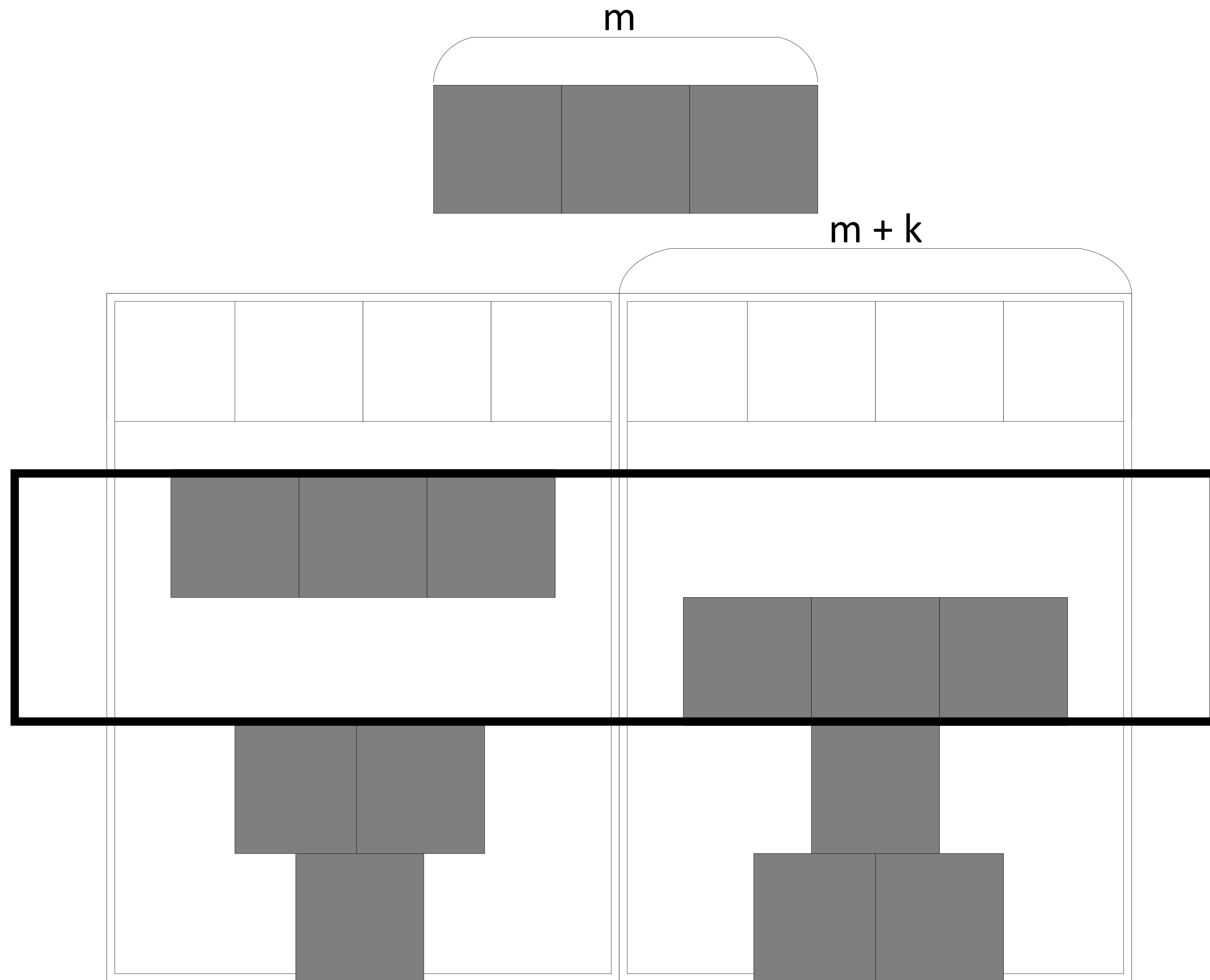
Малишева М.О., група KB-62М

Блок-схема алгоритму послідовного поділу елементів системи на групи для формування базових GL-моделей



Спосіб перетворення GL-моделей за критерієм мінімуму складності реберних функцій
Малишева М.О., група KB-62M

Розподіл відмов зі підмножинами з потужністю $m+k$

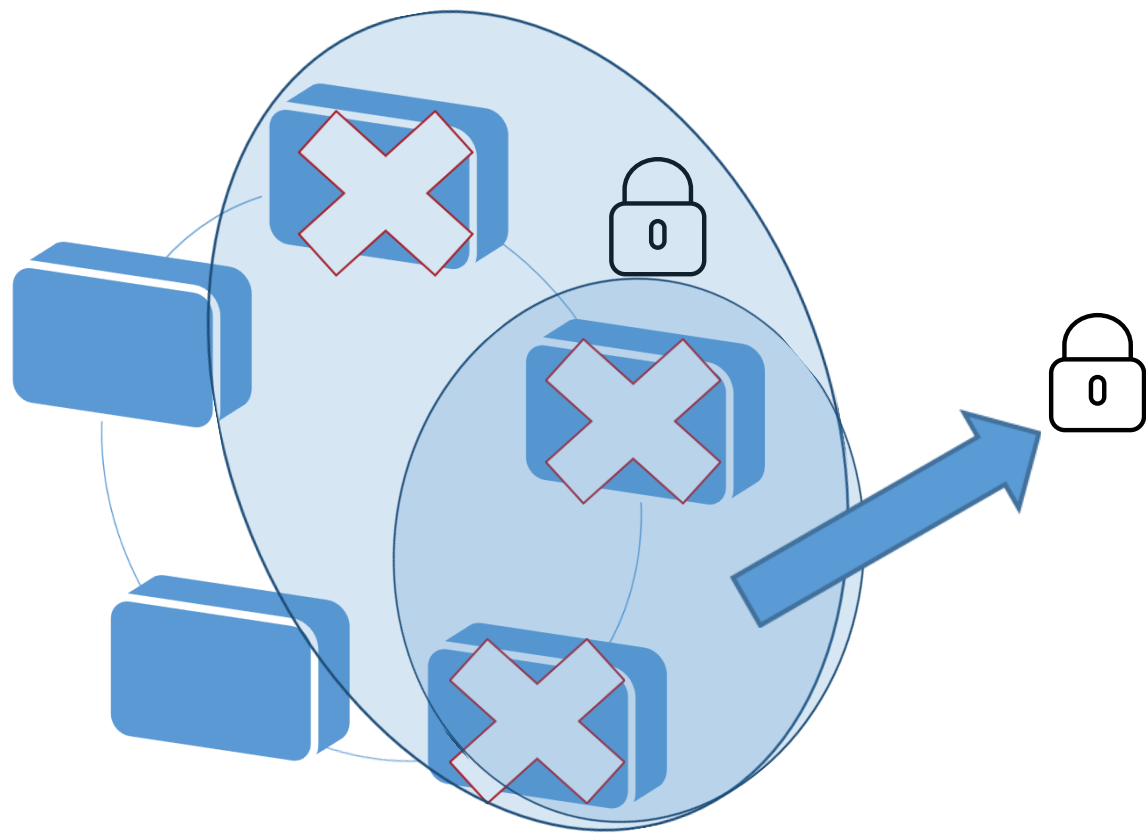


Спосіб перетворення GL-моделей за
критерієм мінімуму складності реберних
функцій
Малишева М.О., група КВ-62М

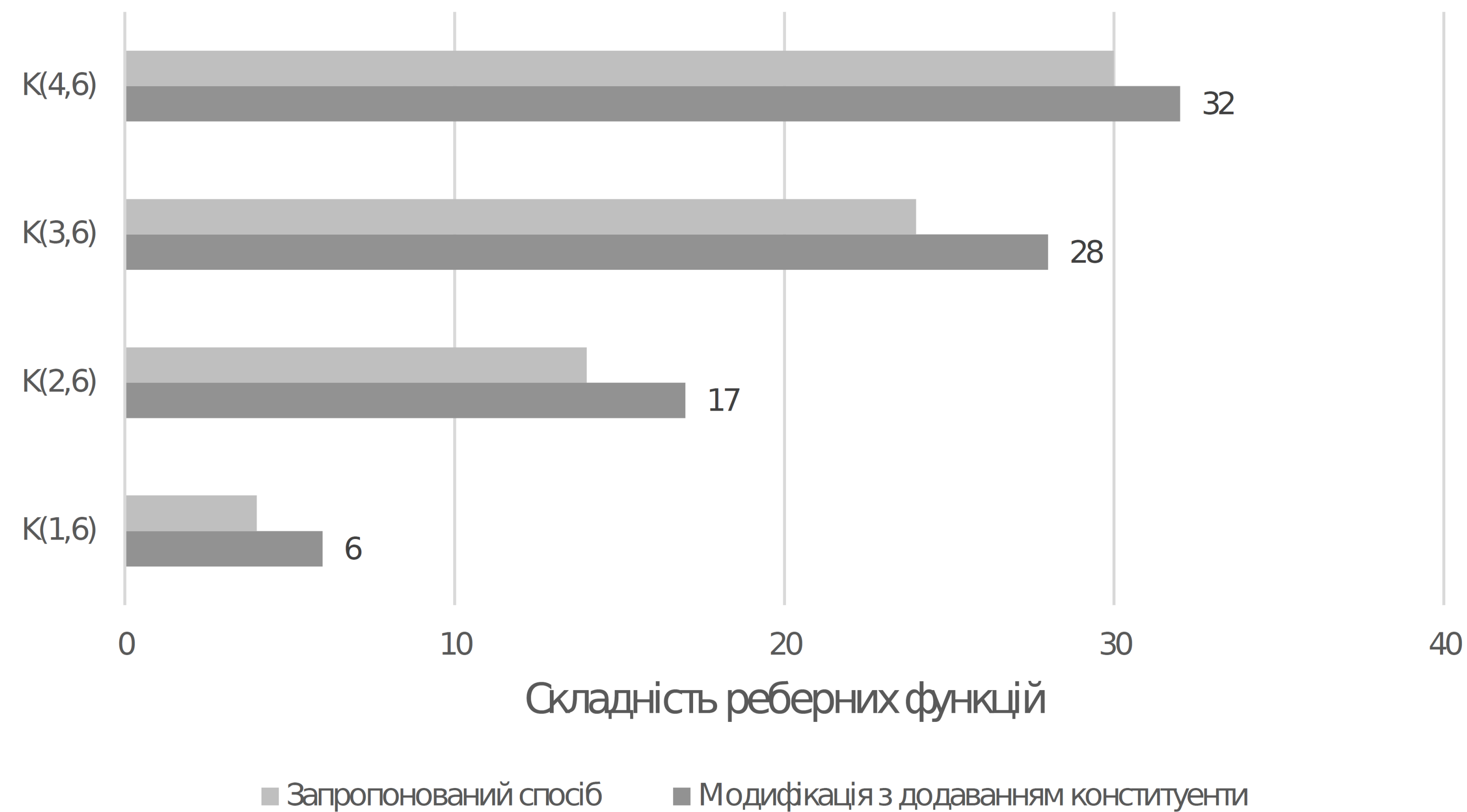
Запропонований спосіб модифікації GL-моделей для відображення підвищення ступеня відмовостійкості системи

$$f_i' = f_i \vee a_{m+k+1} \wedge \dots \wedge a_n$$

b_1	b_2	\dots	b_{m+k}	a_{m+k+1}	\dots	a_n
x	x	x	x	1	1	1

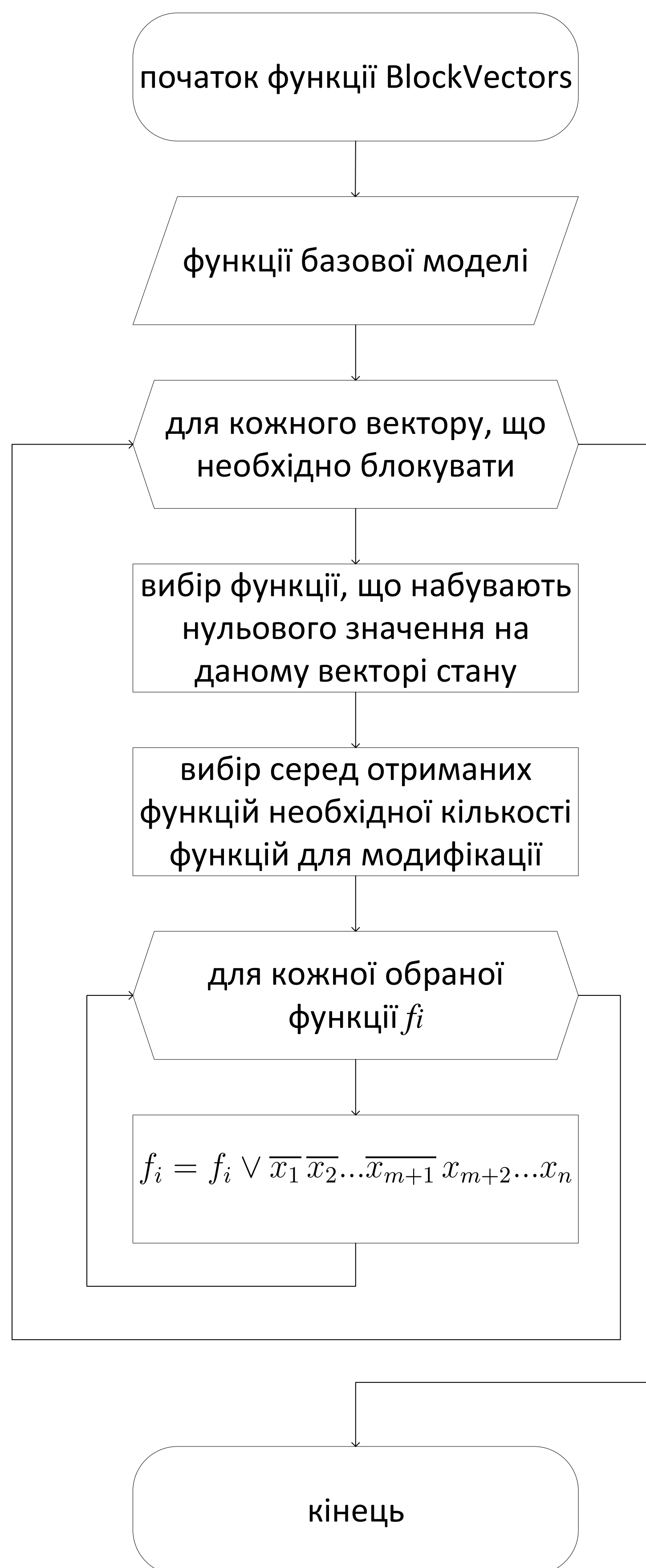
<div><div>$b_1$$b_2$$\dots$$b_{m+1}$$a_{m+k+1}$$\dots$$a_n$</div><div>$x$$x$$x$$x$$1$$1$$1$</div></div>	<div><div>$b_1$$b_2$$\dots$$b_{m+k}$$a_{m+k+1}$$\dots$$a_n$</div><div>$x$$x$$x$$x$$1$$1$$1$</div></div>
<div><div>1) $b_1$$b_2$$\dots$$b_{m+1}$$a_{m+k+1}$$\dots$$a_n$</div><div>$0$$0$$0$$0$$1$$1$$1$</div></div> <div>2) $\leq m$</div>	<div></div>

Аналіз запропонованого способу відображення підвищення ступеня відмовостійкості для блокування векторів з $m+1$ нульовими компонентами



Спосіб перетворення GL-моделей за критерієм мінімуму складності реберних функцій
Малишева М.О., група KB-62M

Відображення підвищення ступеня відмовостійкості системи за допомогою додавання константи



Спосіб перетворення GL-моделей за критерієм мінімуму складності реберних функцій
Малишева М.О., група KB-62М